INSTALACIÓN, CONFIGURACIÓN Y ADMINISTRACIÓN

USERS

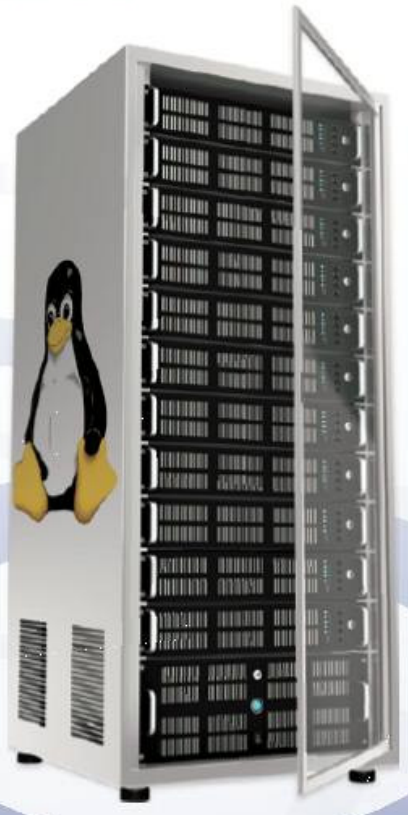
16

Argentina $ 22.- // México $ 49.-

Técnico en REDES & SEGURIDAD

ADMINISTRACIÓN DE SISTEMAS LINUX

En este fascículo veremos cómo configurar y administrar servidores Linux. Conoceremos, además, los comandos más útiles y precauciones de seguridad.



- SERVIDORES BASADOS EN GNU/LINUX
- COMANDOS DE CONSOLA
- LINUX HARDENING
- VERIFICACIÓN DE INTEGRIDAD
- MALWARE EN LINUX

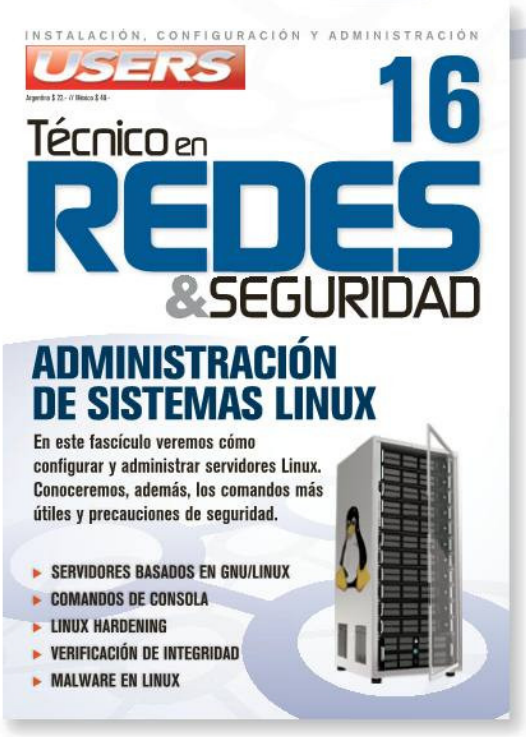
USERS

Técnico en REDES &SEGURIDAD

Coordinador editorial
Paula Budris

Asesores técnicos
Federico Pacheco
Javier Richarte

Nuestros expertos
Valentín Almirón
José Bustos
Gustavo Cardelle
Rodrigo Chávez
Alejandro Gómez
Javier Medina
Gustavo Martín Moglie
Pablo Pagani
Gerardo Pedraza
Ezequiel Sánchez



Impreso en Sevagraf S.A. Impreso en Argentina.

Técnico en redes y seguridad / coordinado por Paula Budris. - 1a ed. - Buenos Aires: Fox Andina, 2013
576 p. ; 28 x 20 cm. (Users; 22)

ISBN 978-987-1857-78-4

1. Informática. 2. Redes. I. Budris, Paula, coord.
CDD 004.68

En esta clase veremos...

Recomendaciones importantes para realizar la administración de sistemas Linux. Conoceremos algunos comandos esenciales y también diversos consejos de seguridad.

En la clase anterior conocimos las principales características de las diversas ediciones de Windows Server y revisamos los conceptos asociados con la asignación de derechos y las restricciones. También vimos qué es Active Directory y aprendimos a administrar las directivas de grupo en forma avanzada. Por otra parte, analizamos la forma de comunicar servidores Linux con clientes Windows y viceversa, y, para terminar, listamos los distintos tipos de malware que existen. En esta clase nos dedicaremos a revisar en detalle la manera en que debemos administrar un sistema Linux. Veremos los comandos de consola básicos y las tareas de Linux Hardening. También realizaremos diagnósticos de red y procesos a través de una consola de comandos, y detallaremos la seguridad a nivel de kernel. Conoceremos sobre los sistemas de verificación de integridad y nos protegeremos contra rootkits, y finalmente, estudiaremos la evolución del malware para Linux y daremos consejos importantes sobre la seguridad en este entorno.

16

6
Comandos de consola

10
Linux Hardening

16
Verificación de integridad

22
Seguridad en entornos de red Linux

Servidores basados en GNU/Linux

Luego de implementar un servidor GNU/Linux, las tareas de administración y configuración iniciales son extensas y variadas; aquí conoceremos cómo realizar las más importantes.

Los sistemas GNU/Linux pertenecen a la familia UNIX y se distribuyen en forma libre, por lo que es posible acceder a su código y modificarlo. Una de las grandes ventajas de la implementación de servidores GNU/Linux es el ahorro en los costos de instalación, pero también se requiere una mayor especialización por parte del personal informático. La puesta en marcha de un servidor basado en GNU/Linux demanda dividir el proceso en varias etapas, de las cuales las más importantes son: **instalación**, **servicios básicos** y **servicios avanzados**.

Gestión de usuarios mediante la consola.

Instalación

Para realizar esta tarea, tenemos que elegir una distribución, de modo que será importante comparar las opciones que nos ofrece el mercado, luego de lo cual iniciamos la instalación mínima y, posteriormente, realizamos el trabajo de configuración de los servicios.

Servicios básicos

La habilitación y configuración de los servicios básicos nos permitirá realizar las tareas más importantes según las necesidades de la red. Por ejemplo, precisamos integrar el quipo en una red, ofrecer un servidor de sitios web con Apache o configurar un servidor FTP; también podemos necesitar funciones de proxy para controlar las conexiones y acelerar la navegación de las PC que se conectan como equipos cliente.

Permisos

Linux es un sistema operativo multiusuario, por lo que debemos ser muy cuidadosos al establecer los permisos para los recursos y usuarios. Los permisos que asignamos a cualquier archivo se componen de tres partes: los permisos del propietario, los permisos del grupo y los permisos del resto. De esta forma, podemos ver que un archivo pertenece a un determinado propietario, a un determinado grupo y, dependiendo de estos permisos, podremos o no acceder a él.

Servicios secundarios

Los servicios secundarios son aquellos que nos permiten, por ejemplo, hacer que el servidor web acepte conexiones internas y también desde Internet, entregue soporte para PHP y CGI, y acepte conexiones por SSH con el fin de administrar la computadora desde cualquier lugar.

Distribuciones

La elección de la distribución que utilizaremos nos demandará algo de tiempo, ya que existen muchas opciones disponibles. En este punto, debemos tener en cuenta diversas características que nos permitirán comparar las ventajas y desventajas de cada una de ellas, hasta tomar la decisión según nuestras necesidades. Los puntos que compararemos en las distribuciones GNU/Linux son los siguientes:

- **Precio**: tengamos en cuenta que, aunque las distribuciones GNU/Linux son libres, no todas se distribuyen en forma gratuita, si bien el precio suele ser menor al que encontramos en otros sistemas operativos, por ejemplo, de la familia Windows. Las distribuciones incluyen un gran número de soportes, los cuales abarcan todos los programas necesarios. Para la mayoría de ellas, como Debian, podemos acceder a su sitio web y descargar todos los CDs o DVDs que corresponden al sistema.
- **Soporte técnico**: en general, las distribuciones GNU/Linux ofrecen un soporte técnico para los usuarios que adquieran el sistema. Si optamos por descargar las imágenes de los discos, tendremos que buscar soporte y ayuda en foros o grupos de usuarios.
- **Versión del kernel**: la versión del kernel o del núcleo del sistema operativo es importante para enfrentar problemas de compatibilidad o de seguridad. Siempre es necesario contar con la última versión disponible. Si elegimos una distribución que no posee un kernel actualizado, tendremos que actualizarlo en forma manual, lo cual es un procedimiento bastante tedioso y no exento de potenciales problemas.
- **Tipo de instalación**: no todas las distribuciones ofrecen ambientes gráficos para realizar la instalación; en algunas de ellas, tendremos que utilizar la consola, razón por la cual, si no somos usuarios expertos, será una buena idea seleccionar una distribución que simplifique el proceso.
- **Gestor de ventanas**: en implementaciones de servidor no utilizaremos mucho el entorno gráfico, pero también podemos instalar algún gestor de ventanas sencillo (como **fluxbox**), para enfrentar eventualidades.
- **Tipo de paquetes**: para instalar aplicaciones, según la distribución, podemos usar distintos tipos de paquetes: **tar.gz** (comprimidos llamados Tarball, contienen el código fuente del programa), **RPM** (se utiliza en forma original para RedHat, pero se ha implementado en otras distribuciones) y **DEB** (formato propio de Debian, también usado por Ubuntu).
- **Otras opciones**: cada distribución está orientada a un público específico, por lo que debemos buscar las opciones adecuadas para implementar un servidor.

La elección de la distribución para nuestro servidor es una tarea personal, pero algunas opciones recomendadas son: **Debian**, **RedHat Enterprise** y **SUSE Linux Enterprise Server**.

SUSE Linux Enterprise: una de las opciones recomendadas para implementar un servidor.

Gestión de usuarios

Una de las tareas más básicas que tendremos que realizar es la gestión de usuarios, para lo cual recurriremos a la consola de comandos. Debemos tener presente que existen dos tipos de usuarios: el administrador, o root; y los usuarios comunes. Para realizar las tareas de administración, necesitaremos poseer un perfil de administrador o root. La administración de usuarios requiere que creemos y gestionemos las cuentas de usuarios, grupos y la asignación de permisos. Esta gestión se realizará cuando debamos establecer políticas de seguridad en el equipo o en la red, o cuando deseemos gestionar servidores NFS, FTP o web. Para comenzar, vamos a gestionar los permisos, mediante el comando ls -l:

```
$ ls -l
total 284
drwxr-xr-x 5 usuario usuario4096
2007-11-26 17:38 2006r3
drwxr-xr-x 5 root root 4096
2007-09-17 15:48 AlberTUX_LIVE
drwxr-xr-x 3 usuario usuario4096
2007-04-02 11:38 Beryl
drwxr-xr-x 2 usuariousuario4096
2007-12-14 15:05 bin
```

La instalación de la distribución Debian puede realizarse mediante un modo gráfico o también un modo de texto.

Las líneas poseen el siguiente formato de estilo:
{T} {rwx} {rwx} {rwx} {N} {usuario} {grupo} {tamaño} {fecha de creación}{nombre}

Campo T: indica qué tipo de archivo es.
Campo {rwx}: permisos que tiene el propietario.
Campo {rwx}: permisos que tiene el grupo.
Campo {rwx}: permisos del resto de usuarios.
Campo {N}: se refiere al número de archivos o de directorios que contiene el elemento.
Campo {usuario}: se trata del nombre del usuario al que pertenece el archivo seleccionado.
Campo {grupo}: nombre del grupo al que pertenece.
Campo {tamaño}: tamaño.
Campo {fecha}: fecha de creación.
Campo {nombre}: nombre

La administración de permisos se realiza mediante la siguiente estructura de comandos:
[chmod] [modo] [permisos] [fichero/s]

Un ejemplo de su uso es el siguiente:
$ chmod -R 755 mi_directorio
$ ls -l
$drwxr-xr-x 2 usuariousuario 4096 2007-07-13 13:57 mi_directorio

También podemos usar los modos para asignar permisos:

a: se aplicará a todos (all)
u: se aplicará al usuario (user)
g: se aplicará al grupo (group)
o: se aplicará a otros (other)
+: se añade el permiso
-: se quita el permiso
r: indica permiso de lectura
w: indica permiso de escritura
x: indica permiso de ejecución

Por ejemplo:
$ chmod -R o-rxmi_directorio

Para agregar un usuario empleamos:
addusernombre_usuario
Adding user 'usuario' ...
Adding new group 'usuario' (1001) ...
Adding new user 'usuario' (1001) with group 'usuario' ...
Creating home directory '/home/usuario' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:

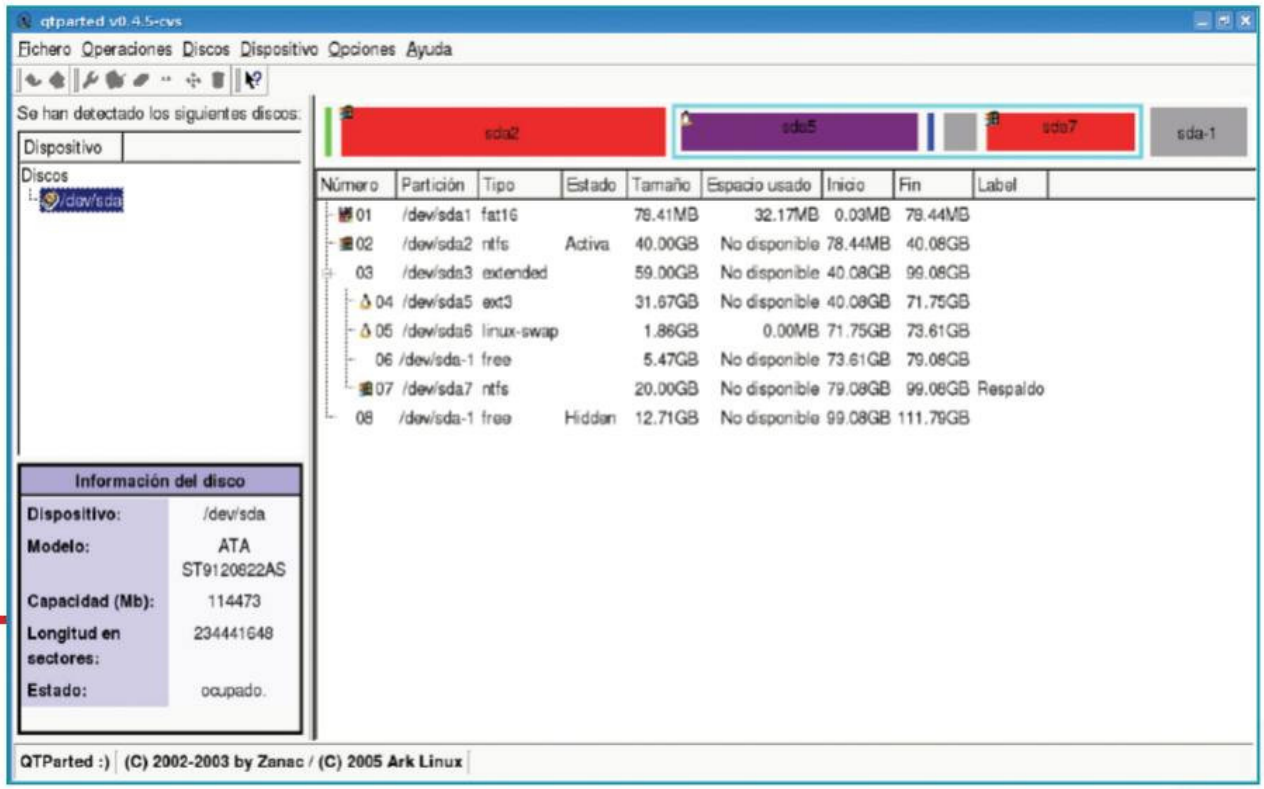
DELUSER NOS PERMITE ELIMINAR UN USUARIO.

Luego de esto, solo ingresamos la contraseña que se asociará al usuario recién creado. La sintaxis completa para la gestión de usuarios es la siguiente:

addusr [-c comentario] [-d home] [-e fecha] [-f dias] [-g grupo] [-G lista de grupos] [-m [-k template] | -M] [-n] [-o] [-p passwd] [-r][-s shell] [-u uid] usuario

Para eliminar un usuario utilizamos:
deluser -R nombre_usuario

La opción -R se encarga de eliminar el directorio home del usuario; sin ella, se eliminará la cuenta de usuario, y quedará el home. Para crear un grupo y asignarle un usuario utilizamos:
groupadd -r NuevoGrupo
gpasswd –a nombre_usuarioNuevoGrupo



Qtparted es una interfaz gráfica que nos permite administrar las particiones de manera sencilla, sin ejecutar comandos.

Gestión de recursos

Sin depender de la distribución de Linux que utilicemos, la gestión de recursos se realiza en forma similar. Entre las tareas que debemos tener en cuenta para administrar los recursos del sistema están mantener las unidades de disco, gestionar el sistema de archivos y controlar los recursos.

Unidades de almacenamiento y particiones

Como sabemos, las unidades de disco están divididas en particiones, las cuales almacenan el sistema que entrega la estructura en la cual se grabarán los archivos. Por ejemplo, el directorio root de un sistema de archivos puede ser montado en cualquier punto del sistema global, aunque generalmente se ubicará en /usr. Una buena idea a la hora de gestionar particiones es utilizar la herramienta fdisk, cuya sintaxis es la siguiente:

```
fdisk [opciones] dispositivo
```

Esta herramienta se utiliza mediante un menú en modo texto, por lo que debemos seguir una serie de pasos tendientes a crear una partición. A continuación, vemos un ejemplo:

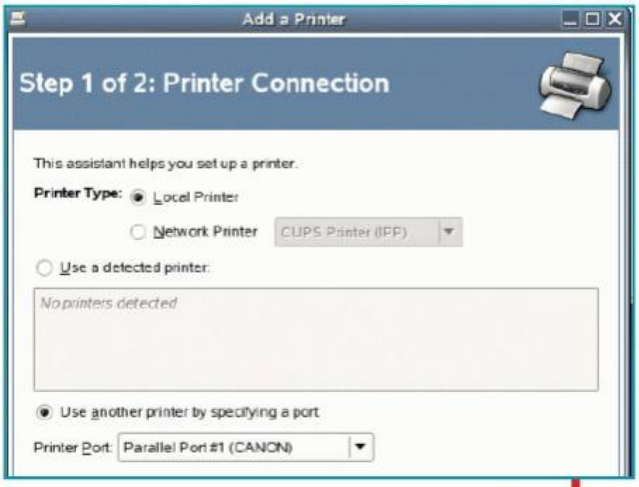
```
Command (m for help): n
Command action
e  extended
p  primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-20805, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-20805, default
20805): +5G

Command (m for help): p

Disk /dev/hdb: 10.7 GB, 10737418240 bytes
16 heads, 63 sectors/track, 20805 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

Device Boot   Start    End   Blocks  Id System
/dev/hdb1       1    9689   4883224+ 83 Linux
```

Otras opciones para gestionar las particiones son las siguientes:

- **cfdisk**: interfaz gráfica para la herramienta fdisk.
- **parted**: programa para crear, destruir, cambiar el tamaño, chequear y copiar particiones en forma sencilla.
- **qtparted**: se trata de una interfaz gráfica de parted, que nos permite manejar particiones sin comandos.

Las tareas de gestión del sistema de archivos son algo complejas y muy extensas, por lo que abordaremos un ejemplo sencillo para crear un sistema de archivos, utilizando el comando mkfs.

```
mkfs [-V] [-t filesystem] dispositivo [n_bloques]
```



La conexión y administración de una impresora puede realizarse mediante asistentes gráficos.

Sus opciones son las siguientes:

- -t filesystem: tipo de sistema de archivos que crearemos.
- n_bloques: número de bloques para el sistema de archivos.

A continuación, vemos algunos ejemplos del uso de mkfs:

- mkfs.ext2 o mke2fs: crea sistemas ext2.
- mkfs.ext3: crea sistemas ext3.
- mkfs.jfs, mkfs.reiserfs, mkfs.xfs: se encarga de crear los sistemas de archivos JFS, ReiserFS y XFS.
- mkfs.msdos, mkfs.vfat: crea sistemas MS-DOS.
- mkswap: crea un sistema de ficheros de intercambio o swap.

Para terminar, es posible realizar el control de los procesos que se ejecutan en el servidor, así como otras tareas de administración importantes. Ahora veremos un listado de las opciones de consola que nos servirán en estas tareas:

```
$ free -m -s 3
```
Muestra el uso de memoria.

```
$ psaux
```
Muestra información de los procesos que están siendo ejecutados.

```
$ top
```
Muestra información de los procesos ejecutados.

```
$ pstree
```
Muestra los procesos, en una estructura de árbol.

```
$ killall proceso
```
Se encarga de detener un proceso.

```
$ strace comando
```
Muestra las llamadas que un proceso ha realizado al sistema.

```
$ fuser -v archivo
```
Muestra los procesos que se encargan de usar un archivo o directorio.

```
$ lsof | less
```
Finalmente, este comando se encarga de mostrarnos el listado de los archivos abiertos por los procesos del sistema. ■

Comandos de consola

Aquí conoceremos los comandos de consola más comunes, que nos servirán para enfrentar diversas acciones en GNU/Linux.

Haciendo una breve reseña histórica, el mundo GNU/Linux comienza en el año 1983, por medio del manifiesto "GNU No es UNIX"; y "Linux", en referencia al creador del Kernel, Linus Torvalds. El objetivo era desarrollar un sistema operativo libre y de espíritu colaborativo. Como primera medida, vamos a familiarizarnos con el intérprete predeterminado de comandos, llamado Bash, presente en la mayoría de las distribuciones GNU/Linux, como Debian y CentOS; y también en algunas versiones de MacOS X y en la mayoría de los sistemas operativos UNIX, ya que es heredado de este último.

Los primeros pasos por consola

Una vez instalado el sistema operativo (sin interfaz gráfica), vamos a iniciar sesión como superusuario o root, ingresando la contraseña definida en la instalación; así ya tendremos listo el prompt para ejecutar comandos. En primera instancia, navegamos con nuestra consola ingresando en directorios del sistema base sobre la partición raíz /, y listando los archivos y directorios que tenemos dentro. Los comandos son los siguientes:

Para ingresar a la raíz del sistema operativo y a sus carpetas:
`root@server:~# cd /`

Podemos visualizar el listado personalizado con los argumentos que nos brinda el comando `ls -ltr`.

```
total 96
drwxr-xr-x   2 root root  4096 2009-12-05 18:55 selinux
drwxr-xr-x   2 root root  4096 2010-04-23 07:11 mnt
drwx------   2 root root 16384 2012-04-11 00:36 lost+found
drwxr-xr-x   2 root root  4096 2012-04-11 00:53 cdrom
drwxr-xr-x   3 root root  4096 2012-04-11 00:55 home
lrwxrwxrwx   1 root root    33 2012-04-11 18:08 initrd.img.old -> boot/initrd.img-2.6.32-40-generic
lrwxrwxrwx   1 root root    30 2012-04-11 18:08 vmlinuz.old -> boot/vmlinuz-2.6.32-40-generic
drwxr-xr-x  16 root root  4096 2012-09-21 11:12 var
drwxr-xr-x  20 root root 12288 2012-12-06 17:10 lib
drwxr-xr-x   2 root root  4096 2012-12-06 17:10 bin
drwxr-xr-x   2 root root  4096 2012-12-06 17:10 sbin
lrwxrwxrwx   1 root root    33 2012-12-06 17:11 initrd.img -> boot/initrd.img-2.6.32-45-generic
lrwxrwxrwx   1 root root    30 2012-12-06 17:11 vmlinuz -> boot/vmlinuz-2.6.32-45-generic
drwxr-xr-x   3 root root  4096 2012-12-30 13:03 media
drwxr-xr-x   3 root root  4096 2013-02-17 20:07 boot
drwxr-xr-x  11 root root  4096 2013-02-18 12:32 usr
drwxr-xr-x   3 root root  4096 2013-02-18 12:32 srv
drwxr-xr-x   3 root root  4096 2013-02-18 13:05 opt
drwx------   6 root root  4096 2013-02-18 16:45 root
drwxr-xr-x  12 root root     0 2013-02-18 16:46 sys
dr-xr-xr-x 124 root root     0 2013-02-18 16:46 proc
drwxr-xr-x  17 root root  4120 2013-02-18 16:46 dev
drwxrwxrwt   7 root root  4096 2013-02-18 16:46 tmp
drwxr-xr-x 136 root root 12288 2013-02-18 16:46 etc
root@server:/#
```

Para listar contenido (directorios o archivos) en este caso, la raíz:
`root@server:~# ls`

Con respecto a ls para listar, podemos utilizar distintos flags o argumentos que permitan visualizar el contenido; por ejemplo:
`root@server:~# ls -ltr`

Donde l es listado extenso, t es ordenar por tiempo de modificación y r es ordenar en forma reversa. Para visualizar todas las opciones de un comando, podemos ayudarnos con el manual incorporado, ejecutándolo de la siguiente manera, en este caso, para ls: `root@server:~# man ls`

AL EJECUTAR EL MANUAL DE TOP, OBTENDREMOS INFORMACIÓN ACERCA DE CÓMO CAMBIAR LA PRIORIDAD DE PROCESOS.

Una vez visualizado, salimos con :q. Luego, para ir ingresando en los demás directorios, también lo hacemos con cd. Cabe recordar que podemos utilizar la tecla TAB para sugerir comandos o completar el contenido: `root@server:~# cd home`
Para volver un nivel: `root@server:~# cd ..`

Comandos de visualización de contenido

Una vez que estemos configurando un servidor, realizando modificaciones o editando algún servicio, debemos listar las configuraciones en cada caso, que se encuentran alojadas, por lo general, en los archivos .conf. Por ejemplo, para el demonio syslog, que administra los registros del sistema en la distribución Debian, lo ubicamos en /etc/rsyslog.conf. Ejecutamos el comando cat (proveniente de concatenar), utilizado para concatenar archivos e imprimirlos por salida estándar (por ejemplo, pantalla). Su sintaxis es muy simple y podemos observarla con man:

```
root@server:~# man cat
root@server:~# cd /etc
root@server:~# cat rsyslog.conf
```

En el caso particular de este archivo, contiene 116 líneas, por lo cual en una pantalla convencional solo podremos visualizar las

últimas, y el inicio quedará sin verse. Para esto, utilizamos el comando less, que nos permite paginar un archivo extenso haciendo que podamos continuar o retroceder tan solo con presionar las flechas del cursor.

```
root@server:~# less /etc/rsyslog.conf
root@server:~# manless
```

Por último, vamos a estudiar el comando **tail**, que proviene de la palabra cola. Este comando es de mucha utilidad porque nos permite visualizar, de manera estándar, las últimas 10 líneas de un archivo. Muchas veces, esto sucede cuando debemos hacer análisis de un log o evento de un servicio y solo nos interesan las últimas líneas, y no queremos utilizar toda la pantalla, ya que puede provocar confusión por la gran cantidad de información que genera un log. Además, con el argumento **-f** (de follow), nos arroja en forma instantánea lo que está sucediendo. Por lo tanto, además de ser un visualizador, también es una herramienta de monitoreo. De aquí nacen otras aplicaciones externas, como **multitail**, que permite ordenar un log por columnas y colores.

```
root@server:~# tail /etc/rsyslog.conf
```

Vamos al ejemplo con argumento incorporado:

```
root@server:~# tail –f /var/log/syslog
root@server:~# mantail
```

Manipulación de contenido

En este ítem, veremos cómo crear, modificar, copiar y borrar contenido desde la consola de un servidor GNU/Linux.
Para crear un directorio en la raíz usaremos el comando mkdir. Primero observamos las características con:

```
root@server:~# man mkdir
```

Luego:

```
root@server:~# mdkir /directorio
```

Ahora, vamos a crear uno dentro de nuestra carpeta de usuario:

```
root@server:~# cd /home/usuario
root@server:~# mkdir directorio
```

Creamos un archivo:

```
root@server: ~# touch archivo01
```

Un comando útil para tener en cuenta es pwd, que nos devuelve en pantalla la ruta en donde estamos ubicados:

```
root@server: ~# pwd
/home/usuario
```

Luego, vamos a modificar este último archivo creado, con mv (abreviatura de move, en inglés):

```
root@server:~# mv archivo01 archivo02
```

Renombramos un directorio o carpeta:

```
root@server:~# mv directorio/ directorio2
```

Navegadores

Los sitios web de la mayoría de los proyectos en GNU/Linux están diseñados para ser visualizados a través de navegadores por consola. Uno de los proyectos más difundidos se llama elinks, y podemos obtenerlo por medio de los repositorios de la distribución Debian o bien del sitio **http://elinks.or.cz/download.html**, para acceder a los fuentes y compilar como vimos anteriormente. Luego, desde la consola ejecutamos, por ejemplo, **elinks www.debian.org**, y ya estaremos listos para navegar desde la shell por el proyecto Debian.

Podemos verificarlo si realizamos un listado de contenido, y veremos la falta del archivo01. Ahora, para poder tener ambos, vamos a copiarlo con copy:

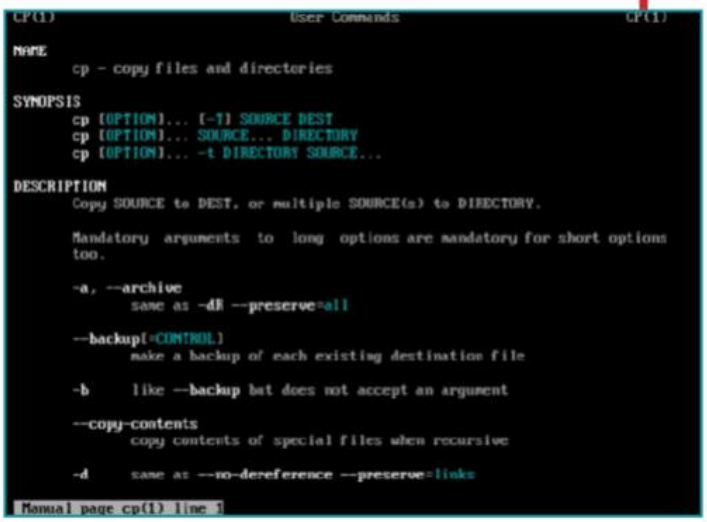
```
root@server:~# cp archivo02 archivo01
```

Para copiar un directorio usamos el flag –r, es decir, copia directorios recursivamente:

```
root@server:~# cp –r /tmp /home/usuario/
```

Es muy útil prestar atención al manual, ya que si aprovechamos el potencial de esta herramienta, ganaremos velocidad en la copia de recursos y también nos será más sencillo programar **scripts**, donde podemos listar el nombre de los recursos copiados en un registro con **--verbose**. Para eliminar contenido, seguimos la misma metodología anterior, pero con rm de remove y -r para

Aquí observamos un ejemplo de la información detallada bajo el comando man de una aplicación. Los administradores de red la usan a diario.



```
#  /etc/rsyslog.conf    Configuration file for rsyslog.
#
#                       For more information see
#                       /usr/share/doc/rsyslog-doc/html/rsyslog_conf.html
#
#  Default logging rules can be found in /etc/rsyslog.d/50-default.conf


#################
#### MODULES ####
#################

$ModLoad imuxsock # provides support for local system logging
$ModLoad imklog   # provides kernel logging support (previously done by rklogd)
#$ModLoad immark  # provides --MARK-- message capability

$KLogPath /proc/kmsg

# provides UDP syslog reception
#$ModLoad imudp
#$UDPServerRun 514

# provides TCP syslog reception
#$ModLoad imtcp
#$InputTCPServerRun 514


###########################
#### GLOBAL DIRECTIVES ####
:
```

Les nos ayudará a visualizar contendido extenso en la consola; con las flechas del cursor, podemos subir y bajar.

directorios: root@server:~# rm/home/usuario/archivo01.
Y para carpetas con: root@server:~# rm-r / /home/usuario/tmp. Aquí eliminamos la carpeta tmp.

Empaquetado y compresión

Llamamos empaquetado a la agrupación de archivos y directorios, lo que da como resultado un solo archivo, pero sin estar comprimido. La herramienta por defecto que tenemos un nuestro sistema operativo GNU/Linux para hacerlo es tar. Para usarla, es necesario combinar diversos argumentos con el fin de indicarle qué es lo que queremos como resultado; por ejemplo, empaquetar o desempaquetar. Vamos a mencionar las variables básicas necesarias:

- -c: crear un nuevo archivo.
- -x: extraer contenido en donde estemos situados en consola.
- -v: indicar que haga una salida de los archivos en el procedimiento.
- -f: permite indicar que el argumento a continuación es el nombre que corresponde al archivo con la extensión .tar.
- -t: argumento para listar contenido y visualizarlo.

```
[root@virtual-XS ~]# df -h
S.ficheros          Tamaño Usado  Disp Uso% Montado en
/dev/sda1             4,0G  2,2G  1,7G  57% /
none                  373M  4,0K  373M   1% /dev/shm
/dev/sdb1             1,8T  518G  1,2T  30% /backup
/opt/xensource/packages/iso/XenCenter.iso
                       52M   52M     0 100% /var/xen/xc-install
[root@virtual-XS ~]#
```

Ejemplo de la aplicación df para visualizar el contenido de nuestras particiones montadas en el sistema; en este caso, un servidor de producción en CentOS.

Por lo tanto, para realizar un empaquetado de nuestro directorio de usuario, es imprescindible no estar situados en consola dentro del directorio por empaquetar; por lo tanto, vamos al raíz:
root@server:~# cd /
root@server:~# tar -cvf backup1.tar /home/usuario

Ahora vamos a listar el contenido de backup.tar:
root@server:~# tar -tf backup1.tar

Por último, vamos a extraer los archivos del archivo backup.tar:
root@server:~# tar -xvf backup1.tar

Luego, tenemos los mecanismos de compresión, que se utilizan para reducir el tamaño de los archivos. Aquí usamos la herramienta **gzip**, que no comprime directorios, sino solo archivos y ficheros creados previamente.
La sintaxis es **gzip+ archivo**, pero podremos incorporar un factor de compresión que va desde 1 a 9, siendo 1 el menor factor de compresión:
root@server:~# gzip -9 backup1.tar

Obtendremos el archivo comprimido backup.tar.gz.
Para descomprimir, utilizamos el argumento -d y volveremos al archivo de tamaño original:
root@server:~# gzip -d backup1.tar.gz

GIT ES UNA HERRAMIENTA USADA PARA DESCARGAR CÓDIGO FUENTE DE APLICACIONES O SINCRONIZAR VERSIONADOS DE ELLAS.

Para finalizar, vamos a utilizar una combinación de empaquetado (tar) y compresión (gzip) dentro de un mismo comando, agregando el argumento -z a la sintaxis:
root@server:~# tar -czfv backup2.tar.gz /home/usuario

Hacemos el proceso inverso para desempaquetar y descomprimir:
root@server:~# tar -xzfv backup2.tar.gz

Estas herramientas son utilizadas, entre otra funciones, para la manipulación de backups y para hacer traslados de estos, ya que facilita la manipulación de volúmenes de información en un solo archivo y reduce su espacio. Además, mediante scripts, podemos agregar valores del tipo fecha a los archivos creados y, así, mantener un orden es nuestra estructura de respaldos.

Utilidades para volúmenes, dispositivos y hardware

Vamos a comenzar con los comandos que nos permitirán visualizar e interpretar los volúmenes, ya sean discos duros, medios extraíbles o unidades de CD-ROM, por ejemplo. Df es una utilidad de reporte sobre el espacio libre en nuestro

sistema de archivos, la cual nos brinda información como espacio total, espacio libre y utilizado, uso en porcentaje y, por último, lugar donde está montado este volumen. Los argumentos básicos o más utilizados son los siguientes:

- h: muestra una visualización sencilla expresada en Megabytes y Gigabytes.
- l: se encarga de brindar la misma información que el argumento anterior, pero expresado en bloques de disco.

Comandos de instalación

En GNU/Linux, tenemos diferentes maneras de realizar instalaciones de paquetes, todo depende de qué distribución estemos usando, en qué formato se encuentren o si tan solo están en los repositorios de la distribución en uso.

Comandos de instalación

Nombre	Distribución	Comando
Deb	Debian GNU/Linux	# dpkg
Apt (*)	Debian GNU/Linux	# apt-get
Rpm	RedHat y derivados	# rpm
Yum (*)	RedHat y derivados	# yum instsall
Ebuild	Gentoo	Compilación del Fuente
Emerge (*)	Gentoo	# emerge
Tar.gz	Todas	Compilación del Fuente

Edición de archivos de configuración

Para modificar archivos de configuración, realizar scripts y hacer tareas de mantenimiento, necesitamos los editores. Distribuciones como Debian GNU/Linux incorporan a nano como editor por defecto, pero la realidad es que el navegador predefinido, presente en la gran mayoría de servidores que configuramos, es **vi**:

```
root@server:~# vi /etc/rsyslog
```

- i: para ingresar en modo inserción de texto y realizar modificaciones.
- o: este comando se encarga de insertar una línea debajo de la actual.
- q: salir sin haber hecho cambios.
- q!: salir sin guardar cambios.
- x: salir guardando los cambios.

Luego, si lo consideramos necesario, podemos instalar, la evolución de la aplicación **vi**, llamada **vim**, la cual presenta grandes funcionalidades, como la posibilidad de abrir varios archivos a la vez y también acceder a la detección de sintaxis en los documentos. ■

Aquí tenemos un ejemplo de búsqueda e instalación de un paquete como SSH con APT, en el sistema Debian GNU/Linux.

```
root@server:~# aptitude install htop iftop
Se instalarán los siguiente paquetes NUEVOS:
  htop iftop
0 paquetes actualizados, 2 nuevos instalados, 0 para eliminar y 0 sin actualizar
.
Necesito descargar 0 B/88,8 kB de ficheros. Después de desempaquetar se usarán 3
03 kB.
Seleccionando el paquete htop previamente no seleccionado.
(Leyendo la base de datos ... 40775 ficheros o directorios instalados actualment
e.)
Desempaquetando htop (de .../archives/htop_0.8.3-1_i386.deb) ...
Seleccionando el paquete iftop previamente no seleccionado.
Desempaquetando iftop (de .../iftop_0.17-16_i386.deb) ...
Procesando disparadores para man-db ...
Procesando disparadores para menu ...
Configurando htop (0.8.3-1) ...
Configurando iftop (0.17-16) ...
Procesando disparadores para menu ...

root@server:~#
```

Procesos en segundo plano

GNU/Linux nos brinda la posibilidad de ejecutar comandos y, luego, enviarlos a segundo plano para seguir trabajando o bien cerrar el shell, por ejemplo, si estamos conectados remotamente por SSH. Para hacerlo, debemos introducir el carácter & al finalizar el comando, de modo de listar los procesos en segundo plano; el comando bg lista los procesos numerados y su estado. Luego, con fg + Numero de Procesos, accedemos al estado de este. Por ejemplo: vi /etc/passwd &, y volvemos a la consola; luego, fg 1, y estamos otra vez en la edición.

Linux Hardening

La seguridad en un servidor muchas veces no tiene la atención que realmente necesita. Aquí conoceremos algunas formas de aumentar el nivel de seguridad en sistemas GNU/Linux.

Estamos acostumbrados a tener en nuestros equipos un antivirus y algún firewall incorporado por el sistema con muy pocas restricciones. Por lo general, y con un poco de suerte, no ocurre nada grave si tenemos esos cuidados, pero dependiendo de la información que manejemos o si estamos conectados a Internet, podemos estar poniendo en serio riesgo nuestro sistema e información al dejar todo librado al azar.

Peligros

Cuando nuestra información es importante o estamos manipulando datos de una empresa, la situación amerita un poco más de responsabilidad y conciencia acerca de los peligros de estar conectados a la red, y no confiar en que los hackers y los virus no van a elegirnos para penetrar en nuestro sistema. Bancos, empresas del gobierno y empresas privadas pagan mucho por protección de sus sistemas informáticos; está más que claro que ellos son los blancos más frecuentes de los "delitos online" debido a la información que manejan. En su mayoría, ejecutan Linux, uno de los sistemas operativos más poderosos y usados para servidores a nivel mundial, tanto por su estabilidad, como por su velocidad. Es uno de los más frecuentes en los equipos empresariales y/o gubernamentales.

```
root@ubuntu: /home/sistemas
        perl besides the one listed in the error message, you may
        override Bastille's search path by setting the
        $CORRECT_PERL_PATH environment variable to the directory
        that the desired perl binary is located in.
        If you don't want to use the default X11 interface then
        run 'bastille -c'. For more information on available interfaces
        see bastille(1m) or run 'bastille -h'

root@ubuntu:/home/sistemas# apt-get install perl-tk
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  linux-headers-3.2.0-29 linux-headers-3.2.0-29-generic
Utilice «apt-get autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
  perl-tk
0 actualizados, 1 se instalarán, 0 para eliminar y 118 no actualizados.
Necesito descargar 2.500 kB de archivos.
Se utilizarán 6.951 kB de espacio de disco adicional después de esta operación.
Des:1 http://archive.ubuntu.com/ubuntu/ precise/universe perl-tk amd64 1:804.029
-1.1ubuntu2 [2.500 kB]
75% [1 perl-tk 1.876 kB/2.500 kB 75%]                          153 kB/s 4seg.
```

La descarga de **Bastille**, y sus dependencias, puede ser realizada desde una consola de comandos.

Configuración

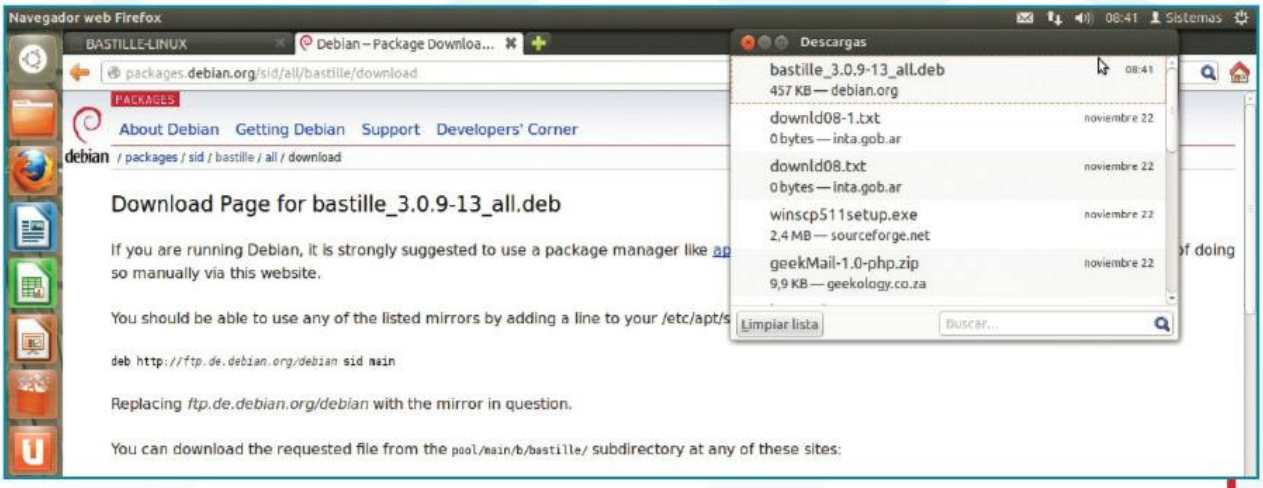
Configurar Linux puede ser una tarea fácil o difícil, dependiendo de cuánto queramos profundizar. Podemos usar el entorno gráfico de algunas versiones, ingresar muchas líneas de comandos, ejecutar scripts o utilizar herramientas que nos ayuden en el proceso de administración. Existen muchas herramientas, como **webmin**, **phpmyadmin** y otras, pero a la hora de configurar parámetros de seguridad, Bastille se destaca por su simpleza y por capacitarnos mientras configura todo lo necesario.

Bastille

Bastille es una herramienta capaz de realizar de forma automática los pasos para configurar varios parámetros de

Bastille

Sabemos que Linux de por sí es seguro, pero debemos tener en cuenta que podemos sellar los agujeros de seguridad que nos queden usando esta herramienta. Su uso es rápido, fácil e intuitivo. Si somos usuarios principiantes, también se nos explica qué hace cada función y por qué conviene asegurarla. Por estas razones, es el software elegido por muchos administradores de red para blindar los datos delicados de las empresas.



seguridad. Fue creada por un grupo de administradores que participaron de la conferencia del Instituto SANS. Jay Beale escribió el programa inicial, y luego, se fueron sumando otros desarrolladores. Por ejemplo, empleados de HP lo insertaron en HP-UX, y trabajadores de IBM ayudaron en Linux Suse y TurboLinux, al igual que una cantidad de voluntarios no remunerados hicieron sus aportes para que Bastille tuviera más opciones y funciones de seguridad.
Podemos obtener este programa ingresando desde nuestro navegador a **http://bastille-linux.sourceforge.net** o ejecutándolo si está en un repositorio del sistema. Al correr Bastille, nos encontraremos con dos modos:

▶ Modo interactivo: de acuerdo con el perfil que elijamos, configurará la fortaleza del equipo. Nos preguntará qué uso vamos a darle y elegirá la opción más adecuada.

▶ Modo manual: nos dejará configurar todos los parámetros de seguridad a nuestro gusto, explicando para qué sirve cada función. Por ejemplo, nos ofrecerá desactivar la parte administrativa solo para algunos usuarios.

BASTILLE ES UNA HERRAMIENTA GRATUITA QUE NOS PERMITE ASEGURAR NUESTRO LINUX.

La seguridad se logra a través de la ejecución de scripts que nos permitirán, por ejemplo, asegurar Apache, instalar SSH, asegurar FTPD, detectar escaneo de puertos, asegurar Send Mail y el booteo de nuestro equipo (por ejemplo, reduciendo el tiempo de espera de LILO y estableciendo una contraseña).
La instalación en un servidor con un sistema Ubuntu es una tarea sencilla porque viene incluido en los repositorios de la distribución y podemos verlo fácilmente por una sesión SSH.
Si lo ejecutamos en modo interactivo, nos hará preguntas para saber qué tipo de uso le daremos al equipo, y nos explicará por qué recomienda configurar ese parámetro.
En caso de usar Bastille en modo texto, utilizaremos los comandos que mencionamos a continuación:

Desde el sitio web de Bastille podemos realizar la descarga de esta aplicación en forma sencilla.

Bastille-Curses-module-1.2.0-1.1mdk.noarch.rpm
perl-Curses-1.05-10.i386.rpm

En tanto que si lo usamos en modo gráfico, recurrimos a:

Bastille-Tk-module-1.2.0-1.1mdk.noarch.rpm
perl-Tk-800.022-11.i386.rpm

Para instalar Bastille ingresamos los siguientes comandos:

```
$ ls
Bastille-1.3.0.tar.bz2
Bastille-Curses-module-1.2.0-1.1mdk.noarch.rpm
perl-Curses-1.05-10.i386.rpm
Bastille-Tk-module-1.2.0-1.1mdk.noarch.rpm
perl-Tk-800.022-11.i386.rpm
$ tarxfj Bastille-1.3.0.tar.bz2
$ rpm -ivh Bastille-Curses-module-1.2.0-1.1mdk.noarch.rpm
$ rpm -ivh perl-Curses-1.05-10.i386.rpm
$ tarxpvf Bastille-1.3.0.tar
$ cdBastille
$ sh Install.sh --> Instala Bastille en las determinadas Paths
$ ./InteractiveBastille --> ejecutar el script
InteractiveBastille
Usage: InteractiveBastille[ -x | -c ] [--norequires]
-x : use the Perl/Tk (X11) GUI
-c : use the Curses (non-X11) GUI
--norequires:askallquestions, evenonesthat do
notapplytothecurrentsystemconfiguration
```

Claramente, podemos apreciar que se dan las opciones tanto para modo texto como para X Windows. Por otra parte, para utilizar el modo texto usamos el comando ./InteractiveBastille –c, mientras que para acceder al modo gráfico, escribimos: el comando ./InteractiveBastille–x. ■

Comandos: diagnóstico de red y procesos

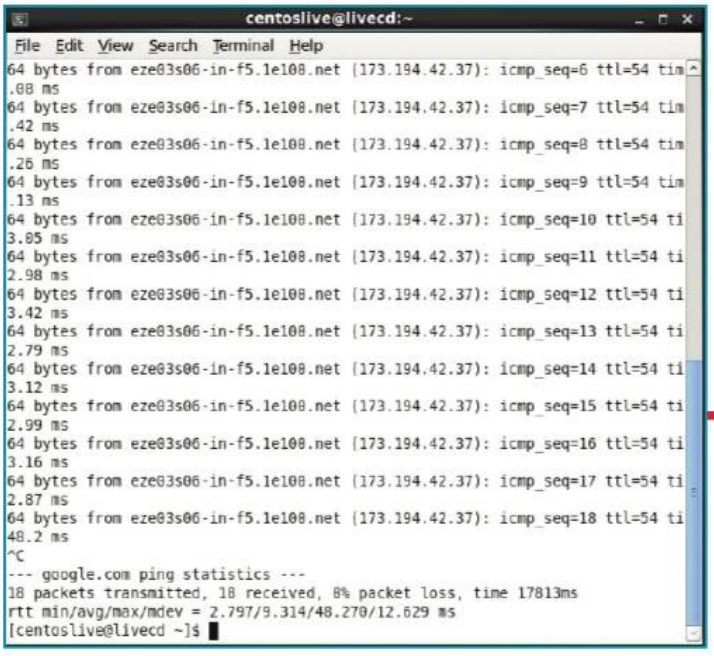
En estas páginas presentaremos un completo listado de los comandos básicos para hacer mantenimiento y localizar fallas en la red.

Dentro del directorio init.d ubicado en /etc o en /etc/rc.d (dependiendo de la distribución), encontraremos una serie de scripts que nos permitirán manipular los servicios instalados en el equipo. La mayoría de ellos reconoce los siguientes argumentos: start, stop, restart y status.
Los nombres de los argumentos describen su función (iniciar, detener, reiniciar y condición), y tienen permisos de ejecución. Siendo root, podremos iniciar un servicio de la siguiente forma:

networking: servicio que controla la tarjeta de red
/etc/init.d/networking start: inicia los servicios de red
/etc/init.d/networking restart: reinicia los servicios de red
/etc/init.d/networking stop: detiene los servicios de red

ifconfig

Este comando nos muestra información sobre la configuración TCP/IP de nuestra computadora: dirección IP, MAC Address, gateway, DNS, etc. Lo utilizamos de la siguiente forma:



Ifconfig: muestra el estado de las interfaces activas.
ifconfig –a: muestra el estado de todas las interfaces, activas o no.
ifconfig ppp0: muestra el estado del ppp0.
ifconfig eth0 up: activa eth0.
ifconfig eth0 down: desactiva eth0.

PARA USAR TCPDUMP, DEBEMOS CUIDAR QUE NUESTRA PC NO QUEDE FUERA DEL UMBRAL DE AUDICIÓN DEL SISTEMA.

Si queremos cambiar dirección IP manualmente, debemos ingresar lo siguiente: ifconfig eth0 [Dirección IP] netmask [máscara subred]. Por ejemplo:

```
ifconfig eth0 192.168.1.102 netmask 255.255.255.0
```

Si nuestra máquina tiene dos placas de red (con una plataforma de VoIPasterisk, por ejemplo), entonces les asignaremos diferentes IP a los distintos ethX. Por ejemplo:

```
ifconfig eth0 192.168.1.102 netmask 255.255.255.0
ifconfig eth1 10.100.0.10 netmask 255.255.0.0
```

En algunas oportunidades, también tendremos que asignarle la dirección de broadcast con la IP del router:

```
ifconfig eth0 192.168.1.102 netmask 255.255.255.0
broadcast 192.168.1.100
```

A diferencia de lo que sucede en sistemas Windows, en GNU/Linux, en forma predeterminada, podemos ejecutar una cantidad infinita de **ping**.

Es recomendable que, después de cambiar una dirección IP, se baje y se suba (reset) la interfaz con los comandos **down** y **up**:

ifconfig down
ifconfig up

iwconfig

Es similar al comando **ifconfig**, pero fue desarrollado para las interfaces wireless. Lo utilizamos de la siguiente forma:

iwconfig: muestra el estado de las interfaces activas
iwconfig eth0: muestra cómo está configurada la placa inalámbrica.
iwconfig ath0: muestra información de la red inalámbrica (nombre de la red, canal, nivel de señal, velocidad, potencia, cifrado WEP, punto de acceso, etc).
iwconfig ath0 essid "Red_WiFI": configura el nombre de la red Wi-Fi o ESSID, con el nombre que queremos asociarnos.

dhclient

Dynamic Host Client se encarga de iniciar la conexión DHCP, mediante el cliente dhcp-client. Usando el parámetro **-r**, liberamos la IP actual; se usa de la siguiente forma:

dhclient eth0 –r: libera la IP actual
dhclient eth0: renueva la IP

netstat

Network Statistics muestra un listado de las conexiones activas, tanto internas (localhost) como externas, los sockets abiertos y las tablas de enrutamiento. Lo usamos de la siguiente forma:

netstat -p: muestra los programas asociados a los sockets abiertos.
netstat -l: muestra los server sockets que están en modo escucha.
netstat -s: muestra información sobre todos los puertos.

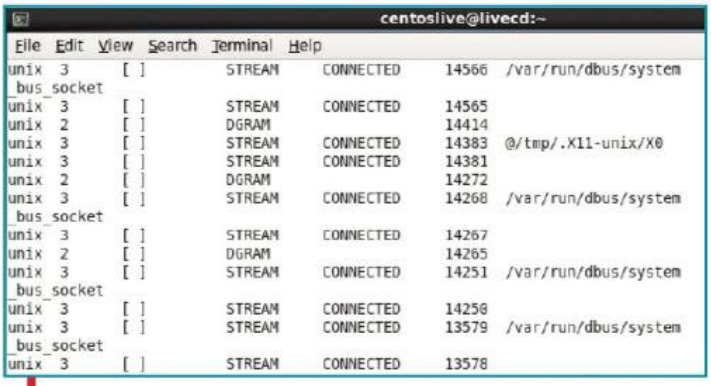
host

Sobre un nombre de dominio, el comando **host** devuelve la IP asociada a él, y viceversa. Sobre una dirección IP, nos devuelve el dominio asociado (DNS lookup). Por ejemplo:
host google.com: muestra la IP de Google
host 8.8.8.8: muestra los DNS públicos de Google

dig

Domain Information Groper o **dig** es una de las mejores opciones a la hora de hacer troubleshooting o debug de problemas DNS. Se usa para obtener una dirección IP a partir del nombre del host (y viceversa), para proveernos de la información de una ruta. Muestra el mapeo de nombres a IP, así como el mapeo inverso de IP a nombres, pero solo sirve para Internet, y no, dentro de nuestra red LAN. Su uso es el siguiente:

dig: realiza una consulta de los NS (Name Servers) raíz.
dig google.com: muestra un registro al DNS de Google.
dig localhost: muestra una respuesta 0, consulta a los DNS del ISP.



Listado de conexiones activas: internas (localhost) y externas.

tcpdump

Es uno de los analizadores de paquetes de red más conocidos, al estilo de **Wireshark** (**www.wireshark.org**). Es un sniffer que monitorea toda la actividad de la red, capaz de escuchar el tráfico de la LAN, y capturar datos para su posterior análisis. A continuación, revisaremos la forma de utilizarlo:

tcpdump –D: muestra la lista de interfaces disponibles.
tcpdump –i wlan0: inicia la captura. La detenemos con **CTRL + C**.

hostname

Nos ayudará a resolver problemas de DNS. El nombre almacenado que identifica cada máquina se encuentra en **/etc/hostname** y podemos consultarlo con **hostname**. Con la variable **files dns**, buscará primero en el fichero **/etc/hosts**, y luego al servidor DNS (en el fichero **/etc/resolv.conf**); su uso es el siguiente:

hostname –f: muestra el nombre y dominio de la PC.
hostname –i: muestra la dirección IP de nuestro nodo.
hostname –a: muestra los alias para nuestro nodo. ■

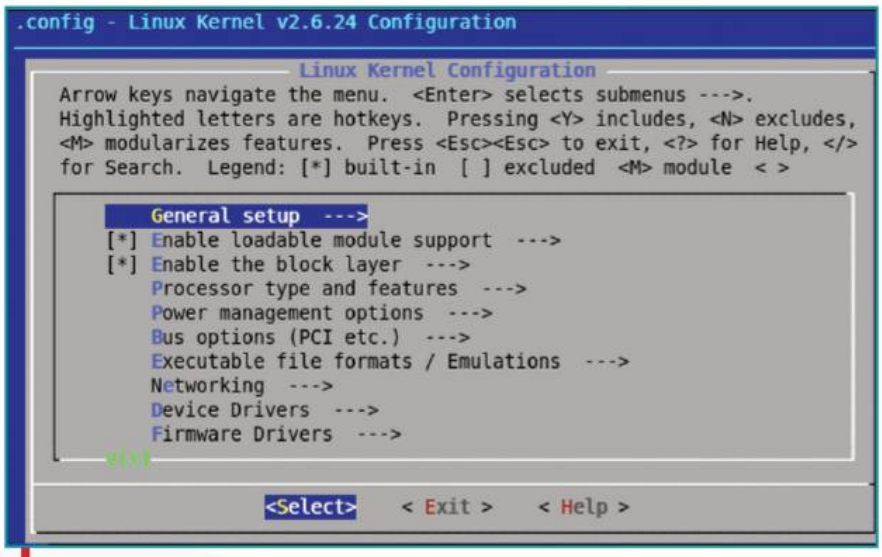
Ping

El conocido ping **(**Packet Internet Groper**) está presente en todos los sistemas operativos y plataformas. Envía paquetes** echo_request **a la dirección IP especificada, para comprobar que la conexión funciona. A diferencia de Windows, en Linux, por defecto, la cantidad de pings es infinita. La variable para limitarlos es** -c**, con la cantidad de paquetes** echo_request **que queremos enviar. Por ejemplo,** ping -c 5 google.com **se encarga de enviar cinco paquetes a** google.com.

Seguridad a nivel de kernel

La seguridad en un sistema puede estar basada en el nivel de aplicación o de kernel; aquí presentaremos este último, analizando sus detalles y la forma en que podemos potenciarlo.

Aquellos que no estén familiarizados con las plataformas **GNU/Linux** o derivadas de **UNIX** quizás encuentren esta sección algo difícil de comprender, pero es importante para todo especialista en informática saber un poco sobre cada uno de los distintos sistemas operativos principales: la rama **Microsoft Windows** y la rama GNU/Linux. Si bien los sistemas del tipo Linux siempre fueron bien vistos en cuanto a la seguridad, lo cierto es que esta seguridad puede dividirse en lo que respecta a sus distintas capas, es decir, desde la más cercana al usuario hasta la más cercana al **hardware**.



Interfaz que permite la configuración de las características del núcleo de Linux.

Seguridad

En la capa donde operan las aplicaciones, Linux provee todo tipo de software orientado a que pueda darse mayor seguridad al sistema desde un uso administrativo, configurando sus características en cuanto a usuarios y contraseñas, controles de acceso, sistema de archivos, inicio y parada, administración general, etc. En este nivel, el usuario administrador deberá ajustar todo haciendo uso de comandos incluidos o no en el sistema, de modo tal que pueda mejorarse la confiabilidad y seguridad total.

Hardening

Si nos encontráramos en sistemas Microsoft Windows, esto sería lo máximo que podríamos hacer como administradores: usar software o

¿Núcleo monolítico o modular?

El sistema GNU/Linux cuenta con un núcleo **monolítico**, es decir que todas las funciones necesarias para él están incluidas en un núcleo único en el que basa su funcionamiento. Su creador en 1991, Linus Torvalds, dio el nombre al núcleo (kernel) y a la plataforma (Linux). A partir de esto, se conoció como GNU/Linux a la combinación de software libre con licencia GNU y un núcleo de Linux. Existen también núcleos que no son monolíticos, y se conocen como **micronúcleos**.

comandos y herramientas del propio sistema que permitan mejorar sus características, o bien adicionarle medidas de control y seguridad para elevar incluso más el nivel de protección. Este proceso es al que nos referimos cuando hablamos de **hardening**, aunque algunos prefieren considerar hardening solo a los ajustes provenientes de los propios comandos y herramientas internas del sistema.
En caso de que queramos realizar ajustes a un mayor nivel de profundidad, no será posible hacerlo en estas plataformas, dado que el sistema se encuentra compilado y funcionando, y no se puede modificar su núcleo ni sus características internas, como el manejo de la memoria, el soporte de características específicas de sistemas de archivos, la forma en que se gestiona el Registro, el uso del disco rígido y el procesador, etcétera.

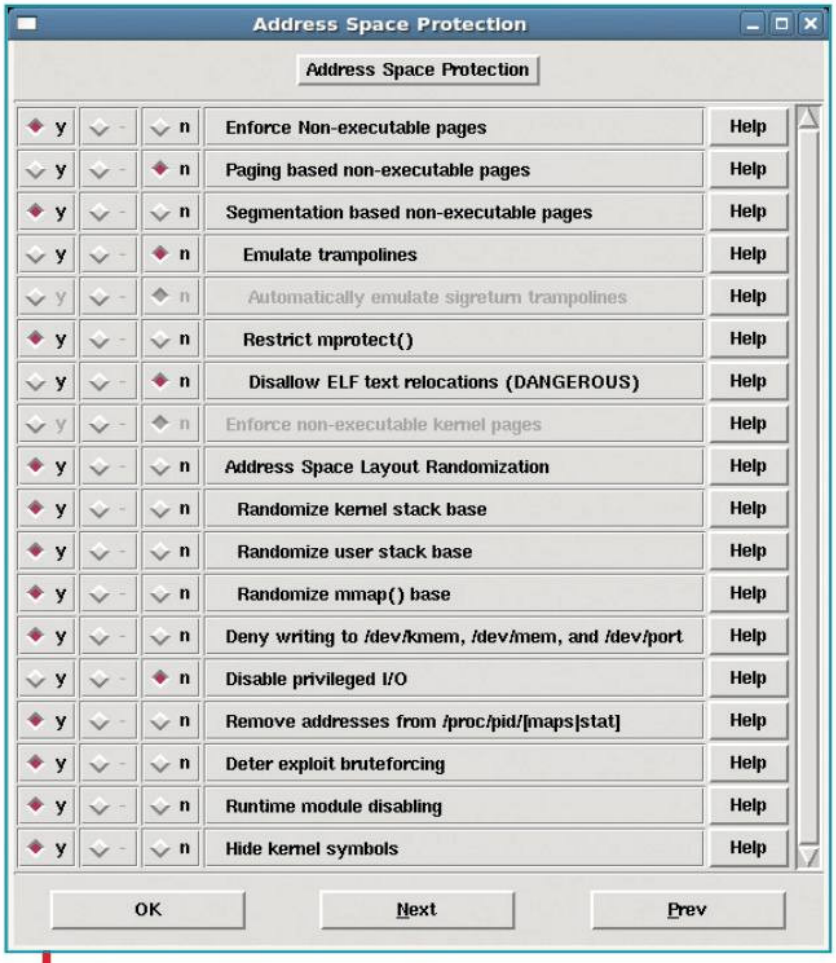
PARA AUMENTAR LA SEGURIDAD DE UN SISTEMA LINUX, NO BASTA CON AJUSTAR SOLAMENTE LAS APLICACIONES.

Mejorar la seguridad

En el caso de sistemas Linux, sí es posible mejorar la seguridad en el nivel del **kernel**, lo cual, por cierto, nos obliga a comprender en mayor profundidad sus distintos aspectos. Para modificar características del núcleo, por empezar, deberemos utilizar alguna herramienta que nos permita gestionar todas las posibles funcionalidades que se pueden cambiar, y luego, pasar por un proceso de recompilación de dicho núcleo a partir del **código fuente**, lo que nos dará como resultado un nuevo núcleo, pero modificado integrando los componentes que nosotros queríamos. Para realizar esta tarea, debemos contar, como dijimos, con el código fuente del núcleo, que obtenemos del sitio oficial **www.kernel.org** o descargándolo mediante el sistema de gestión de paquetes de la distribución que tengamos. Una vez bajado el núcleo sin compilar, debemos bajar alguna interfaz de administración de características para facilitar el acceso a las opciones. Luego, ya podemos navegar por el menú en las diferentes opciones, tanto de funcionalidad como de seguridad, modificar el parámetro al valor que queramos y, finalmente, generar un archivo que corresponda a esos cambios, que luego se aplicarán al código fuente y se volverá a compilar teniendo en cuenta las modificaciones efectuadas.

Características de seguridad

Más allá de las características de seguridad que ya vienen incluidas para cambiar en el núcleo de Linux, también podemos incluir características especiales, que se incorporan de manera directa mediante parches de kernel al código fuente, y agregan más elementos modificables de los que permite en el menú de configuración original con el código original (llamado también **Vanilla**).
Un ejemplo de este tipo de parches es el bien conocido **GRSecurity** (**http://grsecurity.net**), creado para el kernel en su versión 2.4.1, tomando prestados algunos conceptos de **LIDS** y agregando decenas de funciones, como restricción de recursos con alta granularidad, **ACLs** basadas en tiempo y **Role-Based Access Control** (**RBAC**).
Otro sistema de parches para el núcleo de Linux destinado a agregar elementos de configuración de seguridad es el conocido **Openwall** (**Owl**), nacido en 2001 y no tan actualizado como el anterior. Una de las funcionalidades generales agregadas a los parches es el soporte para sistema **PaX**, que refuerza las llamadas al sistema e implementa páginas no ejecutables de memoria, además de aleatorizar el espacio de memoria, para **binarios ELF**. ■

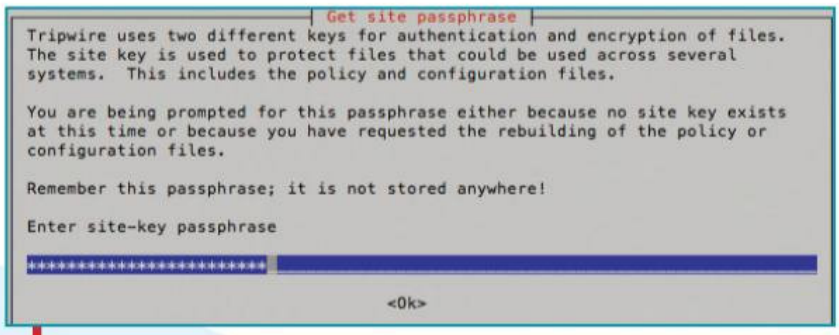


En esta imagen vemos el menú de configuración en modo gráfico que muestra la selección de opciones correspondientes a ASLR.

Verificación de integridad

Es importante establecer un sistema que nos ayude a verificar la integridad de los archivos importantes del sistema; así detectaremos los intentos de modificación malintencionados de dichos archivos.

Sabemos que no existen sistemas completamente invulnerables, razón por la cual es necesario tener en cuenta que siempre estaremos expuestos a la ejecución de diversos ataques. Es cierto que podemos contar con algunas medidas altamente efectivas, tales como firewalls, parches o políticas de control, pero aun en su conjunto, estas no son capaces de brindarnos una seguridad total. Ahora bien, la ejecución de ataques a la red busca ejecutar acciones malintencionadas, tales como modificar en forma parcial los archivos del sistema mediante la alteración o reemplazo de ciertos archivos. Luego, estas modificaciones pueden ser la base para que el atacante tome el control del sistema. Es en este punto donde los sistemas de verificación de integridad cobran una importancia crucial, ya que nos permitirán monitorear los archivos del sistema para asegurar que no sean modificados. A continuación, conoceremos dos de los sistemas de verificación de integridad más utilizados: **Tripwire** y **AFICK**.



En esta ventana ingresamos la contraseña que utilizará Tripwire para autenticar y encriptar los archivos, no debemos olvidarla.

Tripwire

Es una aplicación para entornos Linux, que funciona monitoreando la integridad de aquellos archivos del sistema que son el blanco de ataques. Este sistema de verificación de integridad es capaz de comparar los archivos en los intervalos que configuremos, aunque debemos tener en cuenta que, cuanto más reducidos sean estos intervalos, más recursos del sistema se utilizarán. Obtendremos una copia de esta aplicación Open Source visitando el sitio web oficial en **www.tripwire.org**. Para instalarlo, abrimos una consola de comandos y escribimos los siguientes comandos, presionando la tecla ENTER después de cada uno de ellos.

Para descomprimir el archivo:

```
# tarxvzf tripwire.tar.gz
```

Para instalar la aplicación:

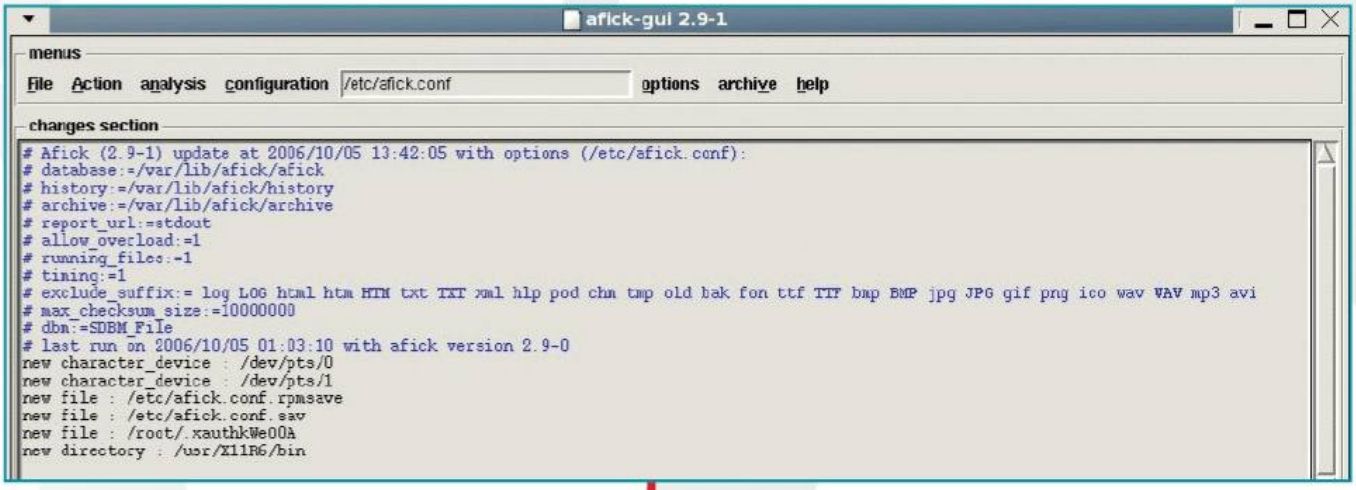
```
# rpm -ivh tripwire-2.3-47.i386.rpm
```

En algunas distribuciones GNU/Linux Tripwire podría encontrarse instalado, por lo que estos pasos no serán necesarios. Una vez que hayamos instalado Tripwire, comenzaremos con el proceso de configuración, para lo cual será necesario definir las claves con el comando:

```
# /etc/tripwire/twinstall.sh
```

Luego de hacerlo, configuramos los archivos del sistema que serán monitoreados por Tripwire. Estos se mantienen en un fichero conocido como **archivo de políticas**. Por suerte para nosotros, Tripwire nos proporciona un archivo que podemos utilizar como plantilla, que se ubica en /etc/tripwire/twpol.txt. Lo abrimos con un editor de texto, como Vi, y buscamos la sección denominada File System and Disk AdministratonPrograms. En ella encontraremos un listado de las ubicaciones que serán monitoreadas; un ejemplo de esta sección es la siguiente:

```
(
rulename = "File System and Disk
AdministratonPrograms",
severity = $(SIG_HI)
)
{
```



Aquí vemos la ventana de ejecución del programa AFICK.

```
/sbin/accton                -> $(SEC_CRIT) ;
/sbin/badblocks             -> $(SEC_CRIT) ;
/sbin/dosfsck               -> $(SEC_CRIT) ;
/sbin/e2fsck                -> $(SEC_CRIT) ;
/sbin/debugfs               -> $(SEC_CRIT) ;
```

Aquí debemos ingresar o eliminar las ubicaciones deseadas. Es importante constatar que la llave de cierre, }, se encuentre al final de las ubicaciones que se monitorearán. Guardamos el archivo y lo instalamos con el comando:

```
# twadmin -m P /etc/tripwire/twpol.txt
```

A continuación, vamos a construir la base de datos en la que Tripwire almacenará el estado actual de los archivos, para lo cual necesitamos ejecutar el comando: `# tripwire -m i 2> /tmp/msj`

Con esto redirigimos los errores a `/tmp/msj`; así podremos revisar los problemas encontrados y corregirlos en el archivo de políticas, para luego instalarlo otra vez y generar la base de datos. Cuando todo se realice sin errores, eliminamos `/tmp/msj`. Cuando terminemos el proceso de configuración de Tripwire, verificamos la integridad del filesystem con el comando:

```
# tripwire -m c
```

AFICK

Afick (**http://afick.sourceforge.net**) es una aplicación que se encarga de supervisar los cambios que se produzcan en el sistema de archivos del equipo, de forma de mantenernos alertados sobre cualquier modificación, lo que podría significar la presencia de una intrusión. Entre las principales características de AFICK encontramos las siguientes:

- Se trata de una aplicación portátil y multiplataforma, por lo que podremos utilizarla en diversos sistemas operativos, tales como Microsoft Windows o GNU/Linux.
- El proceso de instalación de esta herramienta es sencilla, y su inicio y ejecución, bastante rápidos.
- Nos permite acceder a los datos relacionados con los archivos que han sido creados, eliminados o cambiados.
- El archivo de configuración de AFICK nos permite utilizar excepciones y, también, algunos comodines.

En líneas generales, para utilizar AFICK debemos realizar una serie de pasos similares a Tripwire, los cuales incluyen la descarga y ejecución, la edición del archivo de configuración y la creación de la base de datos para comparar en análisis posteriores; luego, hacemos un chequeo del sistema de archivos. A continuación, comentaremos algunos ejemplos del archivo de configuración de AFICK.

Para definir la ruta de la base de datos, usamos el comando:

```
base de datos: = / var / lib / Afick / Afick
```

Para ignorar la estructura del directorio / dev:

```
! / Dev
```

Para excluir todos los archivos o directorios con nombre tmp:

```
exclude_re: = / tmp $ ■
```

Configuración de AFICK

Para editar el archivo de configuración de AFICK, debemos tener en cuenta algunas cuestiones importantes. En primer lugar, distingue entre mayúsculas y minúsculas. Para continuar, tanto en las líneas iniciales como en las finales los espacios en blanco se ignoran. Finalmente, las líneas en blanco o las líneas que comienzan con # serán ignoradas, ya que se consideran comentarios.

Protección ante rootkits

En estas páginas veremos qué son, cómo funcionan, cómo detectarlos, y de qué forma eliminamos los rootkits de sistemas UNIX, Windows, y más. Aprenderemos las mejores prácticas para mantenernos protegidos.

Un **rootkit** es un tipo de malware, es decir, un programa malicioso que se ejecuta sin ser percibido por el usuario o administrador del sistema. Si bien en general se utiliza en servidores, puede afectar cualquier tipo de dispositivo, como computadoras de escritorio, tablets o teléfonos. Implementa un funcionamiento de bajo nivel, haciéndose pasar, por ejemplo, por drivers o componentes del S.O. Su instalación puede ser realizada por un atacante que utiliza una vulnerabilidad o que engaña al usuario, con técnicas de **ingeniería social** o **phishing**. De esta manera, puede ejecutarse durante largos períodos de tiempo sin ser advertido, ya que su objetivo principal no es perjudicar al sistema operativo usurpado, sino servir para otros fines. Los rootkits suelen incluir un backdoor, que permite establecer una conexión remota al sistema. Se los emplea para monitorear el uso del

```
[+] Users, Groups and Authentication
------------------------------------
  - Search administrator accounts...                [ OK ]
  - Checking UIDs...                                [ OK ]
  - Checking chkgrp tool...                         [ FOUND ]
  - Consistency check /etc/group file...            [ OK ]
  - Test group files (grpck)...                     [ OK ]
  - Checking login shells...                        [ WARNING ]
  - Checking non unique group ID's...               [ OK ]
  - Checking non unique group names...              [ OK ]
  - Checking LDAP authentication support            [ NOT ENABLED ]
  - Check /etc/sudoers file                         [ NOT FOUND ]

[ Press [ENTER] to continue, or [CTRL]+C to stop ]


[+] Shells
------------------------------------
  - Checking console TTYs...                        [ WARNING ]
  - Checking shells from /etc/shells...
    Result: found 6 shells (valid shells: 6).

[ Press [ENTER] to continue, or [CTRL]+C to stop ]


[+] File systems
------------------------------------
  - [FreeBSD] Querying UFS mount points (fstab)...  [ OK ]
  - Query swap partitions (fstab)...                [ OK ]
  - Testing swap partitions...                      [ OK ]
```

Comprobaciones realizadas por **lynis**, herramienta desarrollada por Michael Boelen para auditoría de sistemas UNIX.

El caso del rootkit de Sony BMG

En el año 2005 **Mark Russinovich** (cofundador de Sys Internals), utilizando **Rootkit Revealer**, descubrió que Sony BMG incluía en sus CDs un rootkit que modificaba el comportamiento de Windows en cuanto a la reproducción de CDs. Al ejecutar el reproductor de música incluido en el CD, de forma oculta se instalaba el rootkit, que, además, introducía vulnerabilidades en el sistema. Este software fue incluido en gran cantidad de discos y, al darse a conocer la noticia, Sony decidió retirar del mercado todos los CDs con esa tecnología, aunque recibió numerosas demandas en todo el mundo.

sistema, alterar programas, realizar ataques DDoS e IRC, y para enviar spam. Facilitan la formación de botnets, pues es sencillo instalar programas en equipos que tengan un rootkit instalado.

LA MEJOR FORMA DE DETECTAR ROOTKITS ES BOOTEANDO CON UN LIVE DVD, ASÍ NO UTILIZAMOS LOS UTILITARIOS DEL SISTEMA INFECTADO.

Su nombre proviene de los términos *root* (nombre del usuario privilegiado en Linux/UNIX) y *kit* (ya que suelen incluir varias herramientas). En un principio se asociaban solo a sistemas operativos del tipo UNIX, pero también han incursionado en equipos **Windows**, **Mac OSX**, routers Cisco, controladores PLC e incluso en **Android**, **iOS** y **Symbian**.

Recomendaciones

Utilizando buenas prácticas de seguridad, se reduce la posibilidad de ser infectado por un rootkit. Las recomendaciones consisten en:

▶ **Utilizar antivirus**: si bien parece obvio, muchos usuarios y empresas no lo usan o no controlan adecuadamente su correcto funcionamiento y actualización. En grandes ambientes corporativos, es común encontrar equipos cuyo antivirus no fue instalado o no funciona como corresponde. Es preciso desarrollar un proceso que permita detectar con rapidez estas situaciones y corregirlas. A pesar de no ser muy utilizados, también hay antivirus para ambientes UNIX/Linux. Los antivirus más completos permiten detectar la presencia de rootkits.

▶ **Utilizar firewalls**: los firewalls correctamente configurados pueden restringir el tráfico malicioso que genera el malware. Existen dos tipos básicos de firewalls: los de red y los que se ejecutan a nivel del sistema operativo. La adecuada segregación de redes permite realizar una implementación más flexible. Por ejemplo, es recomendable que los administradores y los usuarios finales utilicen distintos segmentos de la red.

▶ **Políticas de password**: la política de password debe requerir contraseñas complejas. Exigir a los usuarios y administradores cambios de contraseña permanentemente puede ser perjudicial: es mejor una clave compleja, que una simple que cambia todos los meses. Cuanto mayor es la cantidad de caracteres y requisitos de complejidad, más difícil es utilizar técnicas de fuerza bruta o ataques por diccionario.

▶ **Mantener el software actualizado**: las actualizaciones de seguridad corrigen fallas de diseño que muchas veces son conocidas y poseen métodos de explotación disponibles públicamente. Es común ver en las empresas que se actualizan solo los equipos con sistema operativo Windows, y no, el resto, como UNIX/Linux. Las aplicaciones también deben ser actualizadas con frecuencia.

▶ **Realizar hardening**: los sistemas operativos suelen incluir muchas herramientas y funcionalidades que posiblemente nunca se utilicen. Es recomendable desactivar todo el software que no va a ser usado y adoptar parámetros seguros para el que sí utilizaremos. Por ejemplo, en sistemas UNIX es aconsejable quitar el conjunto de demonios y utilitarios `rlogin` y configurar `sshd` para que solo utilice la versión 2 del protocolo.

Niveles de ejecución

Existen distintos niveles en los cuales un rootkit puede ejecutarse. Los que se ejecutan en el `ring 3` (modo usuario) corren como una aplicación más. Pueden ocupar el espacio de memoria de otra aplicación para ejecutarse cuando esta es llamada. Los que lo hacen en el `ring 0` (modo kernel) usan drivers o modifican el S.O. para ejecutarse de forma privilegiada. Estos son propensos a generar inestabilidad en el sistema operativo como consecuencia de una programación de muy bajo nivel. Debemos considerar que, al ejecutarse a nivel del kernel, su detección y remoción resultan más complicadas.

Utilidades

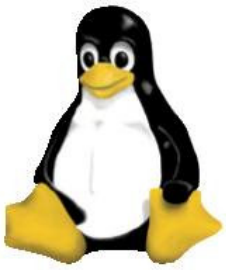
Además de los antivirus tradicionales, existen utilidades especializadas en detectar y remover rootkits conocidos. En la gran mayoría de los casos, permiten revertir las modificaciones que fueron realizadas para pasar desapercibidos.

▶ **chkrootkit** es una herramienta de origen brasileño que permite detectar rootkits instalados en sistemas UNIX/Linux. Consiste en un shell script que se encarga de utilizar herramientas del sistema para detectar las modificaciones realizadas por un rootkit. Puede utilizarse desde un Live CD, que resulta lo más conveniente, ya que de esta manera no utiliza componentes del sistema operativo que puedan estar alterados por malware.

KasperskyTDSSKiller detecta y remueve rootkits conocidos en sistemas Windows analizando archivos del sistema operativo.

▶ **rkhunter** permite identificar rootkits, backdoors y malware en general, comparando el hash de los archivos más importantes del sistema con una base de datos en Internet. Identifica archivos ocultos, permisos incorrectos y cadenas sospechosas en el kernel. Requiere la presencia de comandos como cat, sed, head, tail, stat, readlink o md5/md5sum para realizar las comprobaciones. ■

El malware en los sistemas Linux



Revisaremos la historia y actualidad del malware en los sistemas GNU/Linux, y veremos qué aspectos tener en cuenta y cómo proteger nuestra red de las infecciones provocadas por malware.

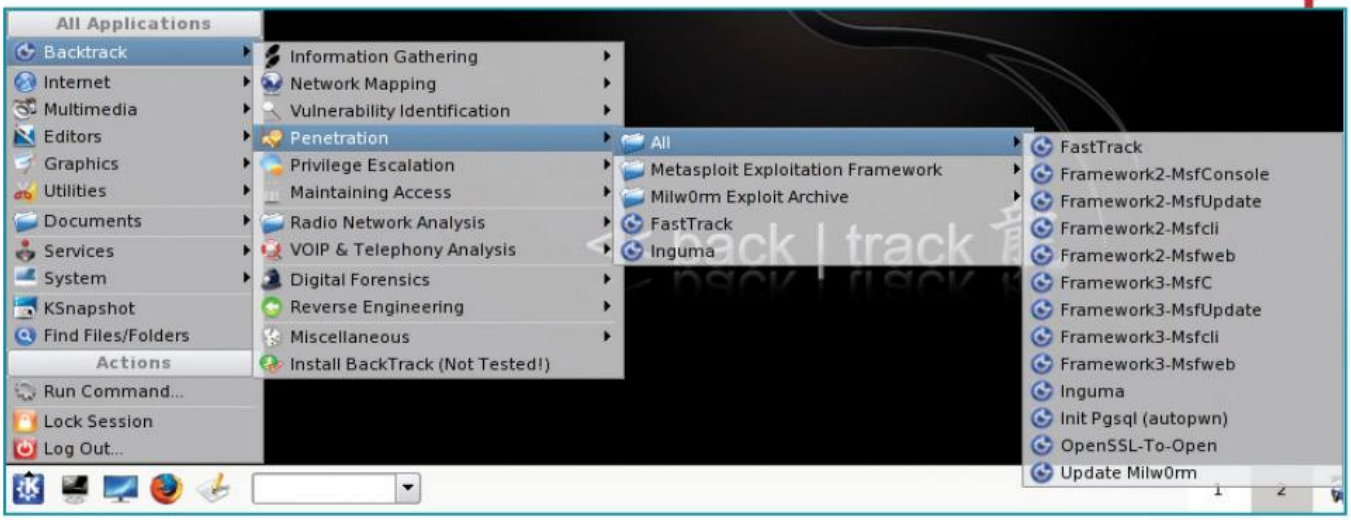
Realmente, es difícil crear malware transportable para equipos GNU/Linux que pueda infectar muchos equipos al mismo tiempo.
Por ejemplo, Apple afirma que su sistema MAC OSX (basado en BSD) no va a quitarnos tiempo con constantes alertas y preguntas de seguridad. Esto es así porque cada Mac viene de fábrica con una configuración segura, de modo que no debemos preocuparnos por entender las complejas configuraciones. Algo similar sucede en el mundo Linux: sabemos que el malware en estos sistemas no es tan común como en Windows. Algunos de los factores que influyen para que esto sea así son los que analizamos a continuación:

- Dependiendo de la distribución, los programas se obtienen de fuentes centralizadas y confiables, y se compilan al momento de realizar el proceso de instalación.
- Los usuarios comúnmente no utilizan el administrador del sistema (root) en forma permanente, como ocurre en Windows. Solo se subroga cuando es necesario. Muchas distribuciones, por defecto, no permiten el login con root, menos aún, en el entorno gráfico (X) utilizado para tareas como navegar por Internet.
- Las configuraciones de seguridad predefinidas de los sistemas GNU/Linux han sido históricamente más seguras que las que encontramos en Microsoft Windows.
- Rápida corrección a las vulnerabilidades detectadas.
- Menor cantidad de usuarios que emplean Linux para desktops.

Malware en GNU/Linux

Como sabemos, es más difícil crear malware para UNIX/Linux, pero aun así, existe, y debemos estar atentos y llevar a cabo algunas acciones para protegernos.
El primer gusano para este sistema fue denominado **gusano de Morris**, en referencia a su creador Robert Morris. Se propagó explotando una vulnerabilidad presente en **Sendmail** (MTA) en el año 1988. Más recientemente, durante los años 2009 y 2010 se detectó malware para GNU/Linux conocido como

La distribución Back Track es importante en el ámbito de la seguridad, por su gran variedad de herramientas preinstaladas.



Consola McAfee ePO, que permite la administración centralizada de los equipos Linux y Windows.

Linux/PsyBot.A. Este era capaz de infectar módems y routers ADSL que utilizaban este sistema operativo y convertirlos en parte del **botnet psyb0t**. Llevaba a cabo ataques **DDoS** (*Distributed Denial Of Service*). También se detectaron troyanos escondidos, por ejemplo, bajo falsos protectores de pantalla, o versiones troyano de software como **Unreal IRC** y **ProFTPD**, este último estuvo activo durante más de seis meses. Uno de los últimos malware detectados para Linux puede automáticamente infectar sitios web de un servidor para atacar a los navegantes con *drive-by-download* (software malicioso que confunde o engaña al usuario para ser instalado e infectar su sistema). Actúa como un rootkit, ocultándose de los administradores. Los navegadores web que accedan a las páginas hosteadas en el servidor infectado serán direccionados a un **iframe** oculto que descargará malware en la computadora cliente. Este malware en particular posee una gran sofisticación, porque se ejecuta a nivel del **kernel**. Por ejemplo, puede infectar servidores en la intranet de una empresa para, luego, infectar a los usuarios internos.

Malware multiplataforma

El malware multiplataforma consiste en un mismo programa que puede ejecutarse e infectar diversos sistemas operativos. Uno de los casos más relevantes fue el troyano **Koobface**, que luego de estar activo durante más de dos años con versiones para Windows, en octubre de 2010 lanzó una primera variante que afectaba también sistemas Linux y Mac usando un applet de Java. Este malware identifica la versión del sistema operativo y ejecuta la amenaza según la plataforma donde reside. El malware que aprovecha el desbordamiento de pila (buffer) se encuentra cada vez más limitado, porque las versiones actuales de kernel para sistemas GNU/Linux protegen el espacio de memoria y de esta forma evitan que los programas puedan infringir el segmento asignado para su ejecución.

Antivirus

Existe una gran cantidad de paquetes antivirus para Linux; algunos de los más conocidos son fabricados por **Symantec**, **McAfee**, **Trend Micro**, **ESET** y **Kaspersky**. También está **ClamAV**, un motor antivirus Open Source con actualización permanente. Este último funciona sobre AIX, BSD, HP-UX, Linux, Mac OS X, OpenVMS, OSF, Solaris e incluso más recientemente en Windows. Las funcionalidades más importantes para tener en cuenta en un antivirus son las siguientes:

- Soporte para la distribución y versión de kernel requerida. Esto es relevante, porque dada la estabilidad de los equipos Linux, muchas veces corren durante años sin necesidad de reinstalación, lo que hace que se continúe utilizando versiones antiguas. Las distribuciones soportadas son SuSe y RedHat, pero algunos soportan además Ubuntu, Debian y Fedora.
- La posibilidad de administrar de manera centralizada los clientes es importante si tenemos muchos equipos; incluso la mayoría utiliza la misma consola que para equipos Windows.
- Porcentaje de detección del total de virus conocidos y no conocidos mediante el uso de tecnologías heurísticas. Existen numerosos proyectos independientes que se encargan de realizan este tipo de análisis.

EL MALWARE MULTIPLATAFORMA IDENTIFICA EL SISTEMA OPERATIVO Y LANZA EL ATAQUE SEGÚN SUS DEBILIDADES.

Dependiendo del rol que cumpla el equipo, deberíamos considerar en mayor o menor medida la utilización de un antivirus. Por ejemplo, un equipo con Samba o shares NFS podría convertirse en un foco infeccioso, sobre todo, para los equipos Windows. Los servidores como SendMail podrían propagar e-mails infectados. También los web servers, como mencionamos anteriormente, podrían infectar a sus clientes. ■

Seguridad en entornos de red Linux

Analizaremos las configuraciones que hacen a la seguridad de Linux, sus componentes más importantes y los valores recomendados.

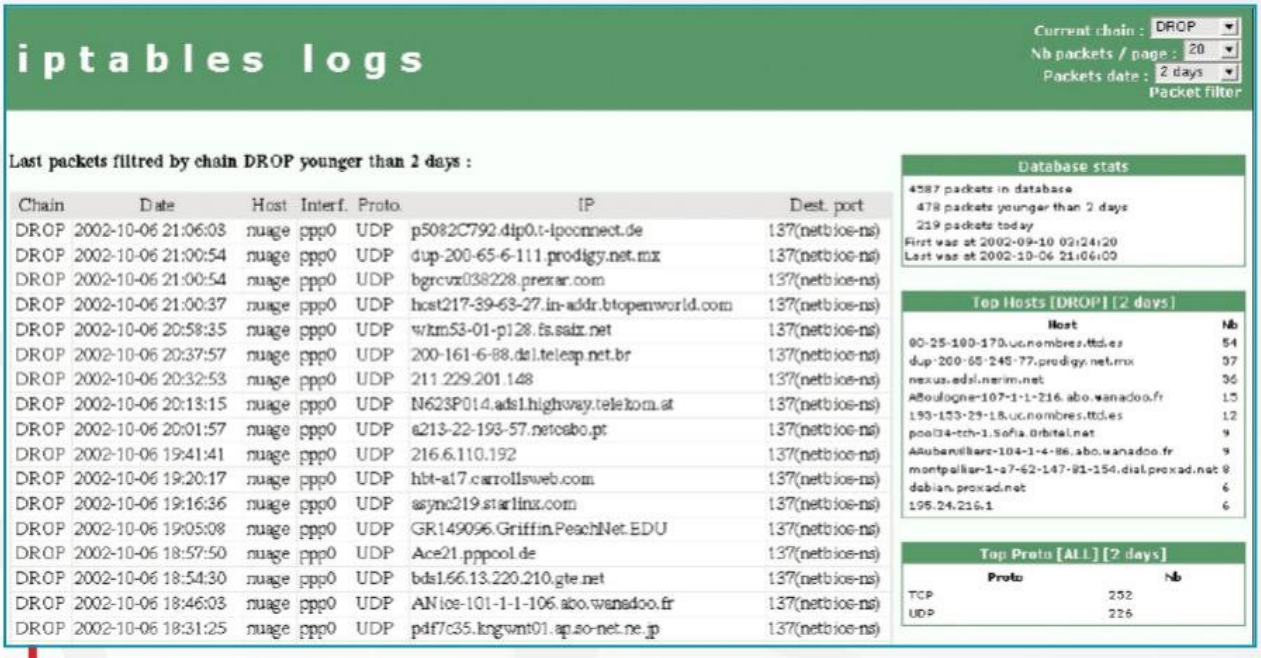
La correcta instalación y configuración de un sistema desde el inicio es fundamental para tener una red segura. La primera buena práctica que debemos tener en cuenta es realizar la instalación básica o mínima que ofrece el instalador Linux. Es decir, no instalar todos los componentes pensando que en algún momento pueden llegar a ser útiles, sino instalar solo lo necesario. Otra buena idea es conseguir las versiones **hardenizadas** (más seguras) de la distribución que vamos a usar. Por ejemplo, **Gentoo** cuenta con una versión **Gentoo hardened**, que puede descargarse directamente. RedHat y SuSe, por su parte, ofrecen una guía de hardening que debe aplicarse luego de la instalación. El **CIS** (*The Center for Internet Security*) es un organismo que desarrolla guías para establecer configuraciones seguras en gran cantidad de sistemas operativos, incluidas muchas distribuciones Linux.

SELinux

Una extensión para Linux muy reconocida en el ambiente de la seguridad es **SELinux** (*Security Enhanced Linux*). Protege el modelo de seguridad tan expuesto para los administradores que posee Linux y brinda una mayor granularidad en los controles de acceso. Por ejemplo, en vez de solo otorgar privilegios de lectura, escritura y ejecución sobre un archivo, permite asignar permisos para anexar, mover un archivo y realizar otras operaciones de manera individual.

Subrogar identidad

Una de las principales ventajas de GNU/Linux es la posibilidad de subrogar identidades (su) como la de root fácilmente, lo que permite trabajar con un usuario no privilegiado y escalar privilegios cuando es necesario. El utilitario SUDO permite la



Iptables puede generar gran cantidad de logs, por lo que existen herramientas específicas para su análisis.

ejecución de comandos con privilegios de otro usuario. Para hacerlo, es preciso definir los comandos permitidos para cada grupo de usuarios. A modo de ejemplo veamos un extracto de sudo.conf:

```
Cmnd_Alias CAUDI = /usr/sbin/aucat,
/usr/sbin/augrep
GAUDI ALL= AUDIT
```

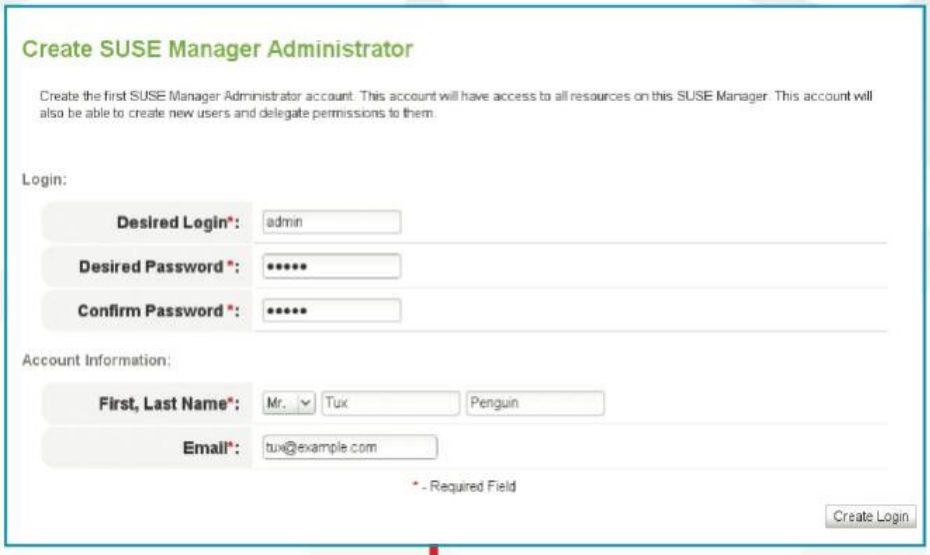
Recomendaciones

En referencia al sistema de archivos, se recomienda montar las particiones de S.O. y los datos usando el sistema **ext3**, que ofrece mayor integridad que las versiones anteriores, gracias a la funcionalidad **journaling**. Además, se recomienda impedir el uso de programas o **shell** con permisos de suid y sgid, con la excepción de los programas predefinidos del sistema u otros requeridos por las aplicaciones específicas, ya que estas funcionalidades permiten que un ejecutable sea invocado con los permisos del owner.

Acceso remoto

En cuanto a los parámetros de acceso remoto y configuraciones de red, consideraremos los siguientes aspectos:

- Deshabilitar **IP-forwarding** a fin de impedir el ruteo de paquetes IP entre las placas de red del servidor. Para esto, ejecutar el comando: # /bin/echo "0" > /proc/sys/net/ipv4/ip_forward.
- Descartar paquetes ping (mensajes ICMP tipo 0), porque podrían brindar información que un atacante fuera capaz de aprovechar.



SUSE Manager es una interfaz web que permite administrar el equipo. Permite administrar los usuarios y la configuración de seguridad.

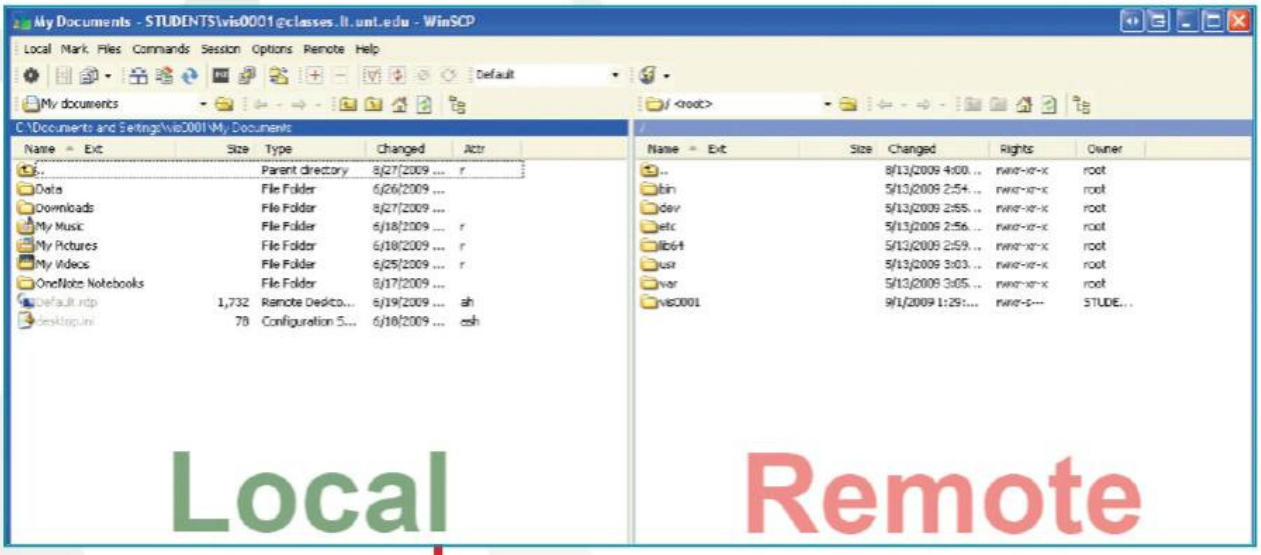
Para hacerlo, ejecutamos el comando: # /bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all.

- Deshabilitar la respuesta a broadcasts ICMP con el fin de no responder a mensajes de ping masivos. Para esto, ejecutar el comando: # /bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts.
- Deshabilitar la aceptación de paquetes enrutados desde el origen. Para esto, ejecutar el comando: # /bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_source_route.
- Deshabilitar la aceptación de paquetes de redirección ICMP, ya que puede alterar tablas de enrutamiento. Para esto, ejecutar los comandos: # /bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_redirects y # /bin/echo "0" > /proc/sys/net/ipv4/conf/all/secure_redirects.
- Habilitar la protección contra respuestas de mensajes de errores falsos. Para esto, ejecutar el comando: # /bin/echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses.

Para garantizar que los usuarios actuales y futuros posean claves de acceso seguras, debemos definir los valores por defecto de las contraseñas que se encuentran en el archivo /etc/login.defs. Todos los usuarios existentes en el sistema deben tener definidos los parámetros de las contraseñas en el archivo /etc/shadow. La estructura es la siguiente:
username : password : lastchg : min : max : warn : expire : disable : reserved
A continuación, conoceremos el significado de cada campo:

Cómo funciona SELinux

SELinux **es una implementación de** MAC **(**Mandatory Access Control**) mediante la cual un administrador del sistemas puede definir cómo las aplicaciones y los usuarios acceden a los recursos (archivos, dispositivos, redes y comunicación entre procesos). Con** SELinux **un administrador puede diferenciar un usuario final, de la aplicación que este ejecuta. Por ejemplo, un usuario puede tener acceso completo a su** home directory**, pero si ejecuta un cliente de correo o un navegador web, este podría no tener acceso, por ejemplo, a las llaves** ssh **que contiene. Es importante considerar que la política centralizada definida le indica al sistema cómo interactúan los componentes.**



WinSCP es un reemplazo seguro para el tradicional intercambio de archivos mediante el protocolo FTP, ya que se basa en SSH.

- Username: identificación del usuario.
- Password: contraseña encriptada.
- Lastchg: días transcurridos desde el último cambio de la contraseña.
- Min: determina la cantidad mínima de días que deben pasar antes que una contraseña pueda ser cambiada.
- Max: determina la cantidad máxima de días que pueden pasar antes que una contraseña deba ser cambiada.

LAS VERSIONES ANTIGUAS DE SSH POSEEN ALGUNAS VULNERABILIDADES.

- Warn: determina la cantidad de días de aviso previos a que la contraseña caduque.
- Expire: determina la cantidad de días que deben transcurrir luego de que la contraseña haya vencido.

El mensaje de preingreso al sistema está ubicado en el archivo /etc/issue. El mensaje de posingreso al sistema se encuentra en el archivo /etc/motd.

Equipos y usuarios

El archivo /etc/hosts.equiv permite definir equipos y/o usuarios sobre los cuales se tendrá confianza, implicando que un usuario pueda iniciar sesión o acceder a recursos en forma remota sin necesidad de brindar una contraseña. Este archivo no debería poseer las entradas del tipo +, ya que permite el acceso desde cualquier host al sistema. La instalación inicial del sistema debería encontrarse libre de servicios de red. Estos deberían ser activados de forma posterior según la necesidad. El servicio de **SSH** se configura en el archivo /etc/ssh/sshd_config. Estos son algunos de sus parámetros:

- ListenAddress: limita las escuchas del protocolo en la dirección definida.
- PermitRootLogin: indica si se permite el ingreso remoto del usuario root.
- PermitEmptyPasswords: determina si impide el acceso de usuarios habilitados sin contraseña asignada.
- AllowGroups sshusers: limita el uso del servicio ssh al grupo sshusers.

Por último, debemos mantener el kernel del sistema y las aplicaciones actualizados. Para esto, recurrimos al comando # yum update o # apt-get update && apt-get upgrade. ■

PRÓXIMA ENTREGA

17

ADMINISTRACIÓN Y ASISTENCIA REMOTA

En el próximo número conoceremos los protocolos necesarios para administrar y ofrecer asistencia a la distancia. También veremos los servicios DDNS y de qué forma podemos hacer uso de las plataformas VNC.

Técnico en REDES &SEGURIDAD

PROFESORES EN LÍNEA
profesor@redusers.com
SERVICIOS PARA LECTORES
usershop@redusers.com

SOBRE LA COLECCIÓN

CURSO VISUAL Y PRÁCTICO QUE APORTA LOS SABERES NECESARIOS PARA FORMAR TÉCNICOS EXPERTOS EN REDES Y SEGURIDAD. INCLUYE UNA GRAN CANTIDAD DE RECURSOS DIDÁCTICOS COMO INFOGRAFÍAS, GUÍAS VISUALES Y PROCEDIMIENTOS REALIZADOS PASO A PASO.

Con la mejor metodología para llevar adelante el montaje y mantenimiento de las redes informáticas y con los aspectos clave para brindarles la protección necesaria, esta obra es ideal para aquellos aficionados que deseen profundizar sus conocimientos y para quienes quieran profesionalizar su actividad.

REDES &SEGURIDAD 00016
9 789871 857784

CONTENIDO DE LA OBRA