



Capítulo 25

Seguridad de Linux

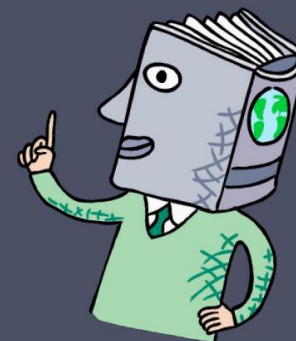
Contribuido por:
Mick Bauer

Editor de seguridad, Linux Journal

Director de servicios con valor sustraído, Wiremonkeys.org


Linux

- Creado en 1991 por Linus Torvalds
- Se ha convertido en uno de los sistemas operativos más populares y versátiles del mundo.
 - o Gratis
 - o De código abierto
 - o Disponible en una amplia variedad de distribuciones dirigidas a casi todos los escenarios de uso imaginables
- Algunos ejemplos de distribuciones incluyen:
 - o Red Hat Enterprise Linux
 - Conservador y con apoyo comercial.
 - o ubuntu
 - Completamente gratis
 - o uClinux
 - Versión integrada simplificada pero hiperestable diseñada para su uso en electrodomésticos y productos de consumo



Seguridad de Linux

El modelo de seguridad tradicional de Linux se puede resumir de forma bastante sucinta: las personas o los procesos con privilegios de “root” pueden hacer cualquier cosa; otras cuentas pueden hacer mucho menos.




Desde la perspectiva del atacante, el desafío de hackear un sistema Linux es obtener privilegios de root.



Una vez que un atacante obtiene privilegios de root, puede:

- Borrar o editar registros
- Ocultar sus procesos, archivos y directorios
- Redefinir básicamente la realidad del sistema tal como la experimentan sus administradores y usuarios



Por lo tanto, la seguridad de Linux (y la seguridad de UNIX en general) es un juego de “el root se lo lleva todo”.

Acceso discrecional

Controles (DAC)

El modelo
de seguridad
de Linux

En Linux
Sistema DAC
hay

- Usuarios: cada uno de los cuales pertenece a uno o más grupos
- Objetos: archivos y directorios

Los usuarios
leen, escriben
y ejecutan los
objetos
según los
permisos del
objeto.

Cada objeto
tiene tres
conjuntos
de permisos:

- Usuario-propietario
- Propietario del grupo
- Otros
(todos los demás)

Permisos
son
aplicados por el
kernel de Linux

Permisos



Antes de ejecutarse, los permisos de archivo de un programa restringen quién puede ejecutarlo, acceder a él o modificarlo.

Cuando se ejecuta, un proceso normalmente se ejecuta con la identidad del usuario y el grupo de la persona o proceso que lo ejecutó.

Si un proceso en ejecución intenta leer, escribir o ejecutar algún otro objeto, el núcleo primero evaluará los permisos de ese objeto contra la identidad del usuario y el grupo del proceso.

Quien sea propietario de un objeto puede establecer o cambiar sus permisos.

La **cuenta de superusuario del sistema** tiene la capacidad de tomar posesión y cambiar los permisos de todos los objetos del sistema.

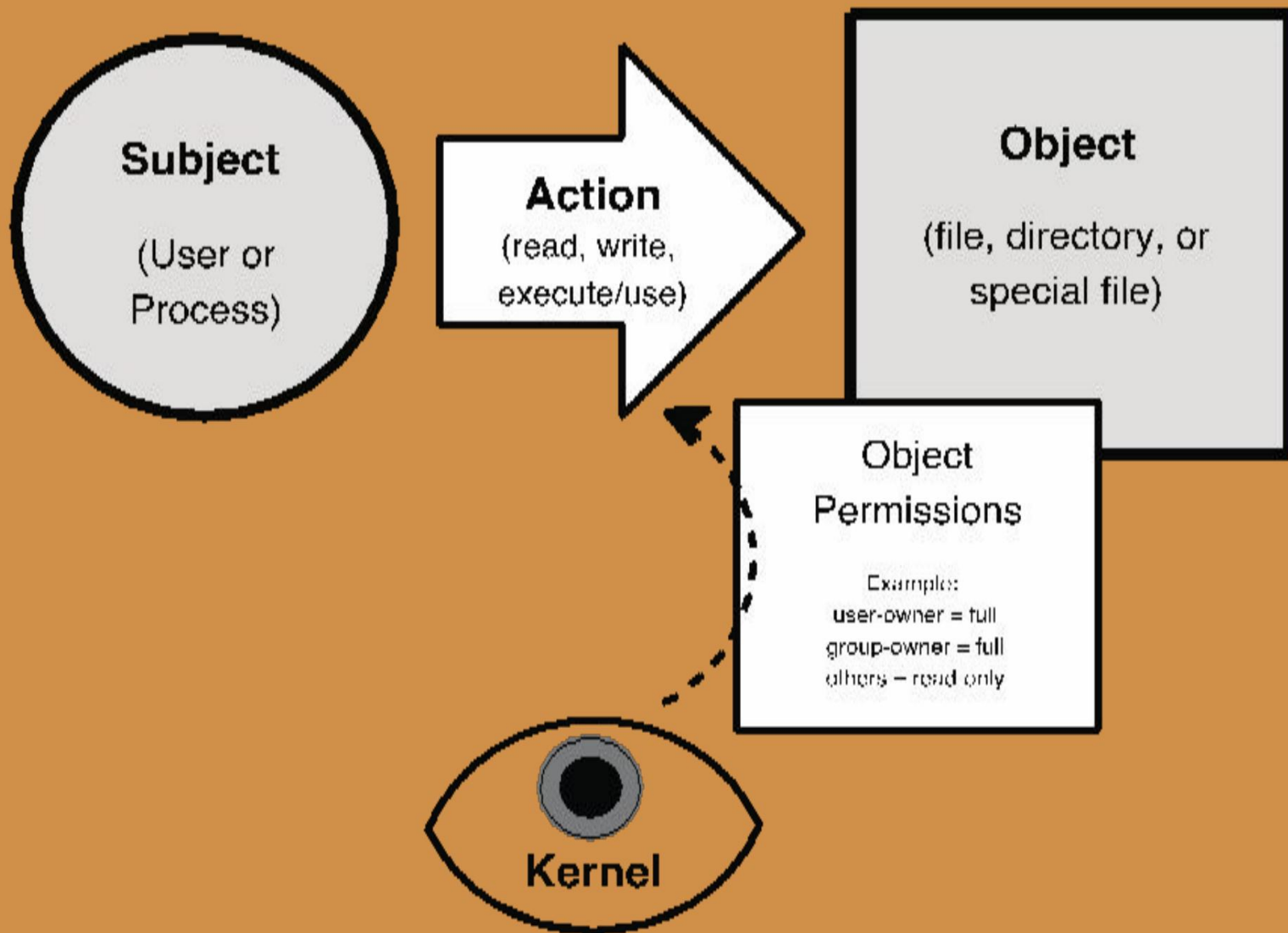


Figure 25.1 Linux Security Transactions

Convertidor CA

En cierto sentido, Linux trata todo como un archivo.

En UNIX, un directorio es en realidad un archivo que contiene una lista de otros archivos.

Para el kernel de Linux, la unidad de CD-ROM conectada a su sistema también es un archivo (`/dev/cdrom`)

- Para enviar datos hacia y desde la unidad de CD-ROM, el núcleo de Linux lee y escribe desde este archivo.

Otros archivos especiales, como las tuberías con nombre, actúan como entrada/salida. Conductos (E/S) que permiten que un proceso o programa pase datos a otro

- Un ejemplo de una canalización con nombre en sistemas Linux es `/dev/urandom` (devuelve caracteres aleatorios del generador de números aleatorios del núcleo)

Usuarios, grupos y Permisos

- Hay dos cosas en un sistema UNIX que no están representadas por archivos:
 - o Cuentas de usuario
 - o Cuentas de grupo
- Cuenta de usuario
 - o Representa a alguien o algo capaz de utilizar archivos.
 - o Puede asociarse tanto con seres humanos reales como con procesos
- Cuenta de grupo
 - o Una lista de cuentas de usuario
 - o Cada cuenta de usuario se define con una membresía de grupo principal, pero puede pertenecer a tantos grupos como necesite.

Usuarios, grupos y Permisos

```
maestro:x:200:100:Maestro Edward Hizzersands:/inicio/maestro:/bin/bash
```

Listado 25-1: Una entrada `/etc/passwd` para el usuario "maestro"

```
directores:x:100:
```

```
pianistas:x:102:maestro,volodya
```

Listado 25-2: Dos entradas `/etc/group`



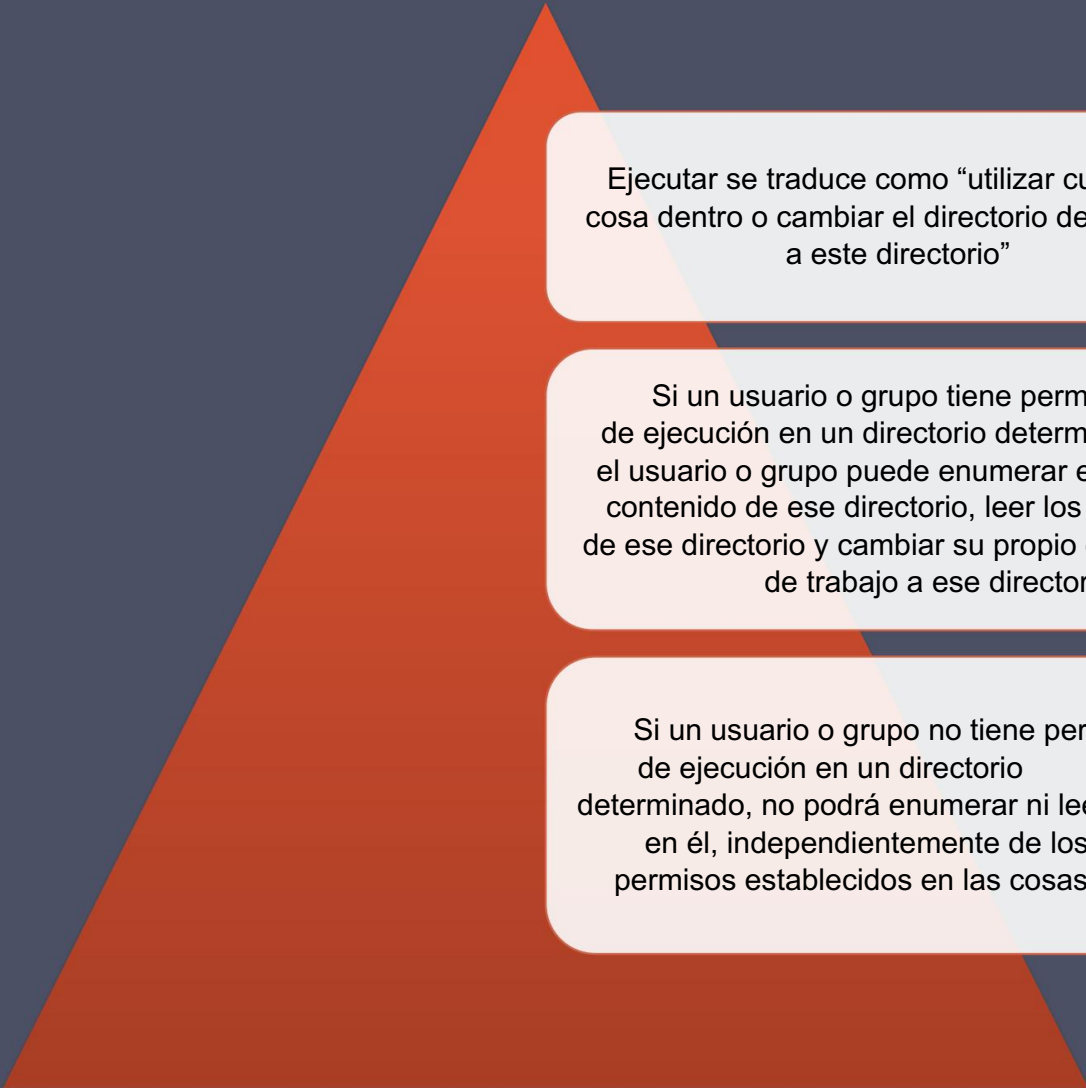
Permisos de archivos simples

- Cada archivo en un sistema UNIX tiene dos propietarios (un usuario y un grupo)
- Cada usuario y grupo tiene su propio conjunto de permisos. que especifican lo que el usuario o grupo puede hacer con el archivo (leerlo, escribir en él, eliminarlo, ejecutarlo)
- Otro
 - o Cuentas de usuario que no poseen el archivo o pertenecen al grupo que es dueño de ello
- El listado 25-3 muestra una lista larga de archivos para el archivo /home/maestro/baton_dealers.txt

```
-rw-rw-r-- 1 maestro directores 35414 25 mar 01:38 baton_dealers.txt
```

Listado 25-3: Listado de archivos que muestra permisos

Permisos de directorio



Ejecutar se traduce como “utilizar cualquier cosa dentro o cambiar el directorio de trabajo a este directorio”

Si un usuario o grupo tiene permisos de ejecución en un directorio determinado, el usuario o grupo puede enumerar el contenido de ese directorio, leer los archivos de ese directorio y cambiar su propio directorio de trabajo a ese directorio.

Si un usuario o grupo no tiene permisos de ejecución en un directorio determinado, no podrá enumerar ni leer nada en él, independientemente de los permisos establecidos en las cosas dentro.

Listado 25-4:

Un directorio legible por grupos

```
bash-$ chmod g+rx extreme_casseroles
```

```
bash-$ ls -l extreme_casseroles
```

```
drwxr-x--- 8 bateristas biff 288 25 de marzo 01:38 extreme_casseroles
```

La parte pegajosa

- En los sistemas operativos UNIX más antiguos, el bit adhesivo era
Se utiliza para escribir un archivo (programa) en la memoria para que se cargue más rápidamente cuando se invoca.
- El bit adhesivo se utiliza para permitir que alguien con permisos de "escritura" cree nuevos archivos en el directorio pero no elimine ningún archivo.
- En Linux, cuando se configura el bit persistente en un directorio, se limita la capacidad de los usuarios de eliminar cosas en ese directorio.
 - o Para eliminar un archivo determinado en el directorio, debe ser propietario de ese archivo o poseer el directorio, incluso si pertenece al grupo propietario del directorio y tiene permisos de escritura grupales configurados en él
- Para configurar el bit fijo, emita el comando:

`chmod +t nombre_directorio`

Listado 25-5:

Contenido de extreme_casseroles/

drwxrwxr-T 3 bateristas biff 192 2004-08-10 23:39 .

drwxr-xr-x 3 bateristas biff 4096 2004-08-10 23:39 ..

-rw-rw-r-- 1 bateristas de biff 18 2004-07-08 07:40

cazuela_de_pavo_con_chocolate.txt

-rw-rw-r-- 1 bateristas biff 12 2004-08-08 15:10

pineapple_mushroom_sorpresa.txt

drwxr-xr-x 2 bateristas geniales 80 10-08-2004 23:28 fuente

Listado 25-6:

Intentando eliminar con el bit fijo configurado

(entrada del usuario en negrita)

```
crash@localhost:/extreme_casseroles> rm  
piña_champiñón_sorpresa.txt
```

```
rm: no se puede eliminar 'pineapple_mushroom_suprise.txt': Operación no  
permitida
```

Setuid y Setgid

Dos de los bits de permisos más peligrosos en UNIX

Si se configura en un archivo binario ejecutable, el bit setuid hace que ese programa se ejecute como su propietario sin importar quién lo ejecute.

Si se configura en un ejecutable, el bit setgid hace que ese programa se ejecute como miembro del grupo que lo posee, independientemente de quién lo ejecute.

Muy peligroso si se configura en cualquier archivo propiedad de root o cualquier otra cuenta o grupo privilegiado

Setgid y directorios

Setuid no tiene efecto sobre los directorios, pero setgid sí.

Establecer el bit setgid de un directorio hace que cualquier archivo creado en ese directorio herede el grupo propietario del directorio.

- Esto es útil si los usuarios de su sistema tienden a pertenecer a grupos secundarios y rutinariamente crean archivos que necesitan ser compartidos con otros miembros de esos grupos.

Si el directorio no es escribible por el grupo, el bit setgid no tendrá efecto porque los miembros del grupo no podrán crear archivos dentro de él.

Modos numéricos



- Internamente, Linux utiliza números para representar permisos.
- Consta de cuatro dígitos
 - o A medida que lee de izquierda a derecha, estos representan permisos especiales, permisos de usuario, permisos de grupo y otros permisos.
- Cada permiso tiene un valor numérico y los permisos en cada lugar de dígito son aditivos.
 - o El dígito representa la suma de todos los bits de permiso que desea configurar
- Los valores numéricos básicos son 4 para lectura, 2 para escritura y 1 para ejecutar
 - o Estos valores representan bits en un flujo binario y, por lo tanto, son todos potencias de 2
 - o Si los permisos de usuario se establecen en "7", esto representa 4 (valor para lectura) más 2 (el valor para escribir y 1 (el valor para ejecutar)

Especial Permisos

4 significa setuid,
2 significa setgid,
1 significa bit pegajoso

Por ejemplo, el modo
numérico 3000 se traduce
a

"setgid establecido, sticky-bit
establecido, ningún
otro permiso establecido"

(que es, en realidad, un conjunto
de permisos inútil)

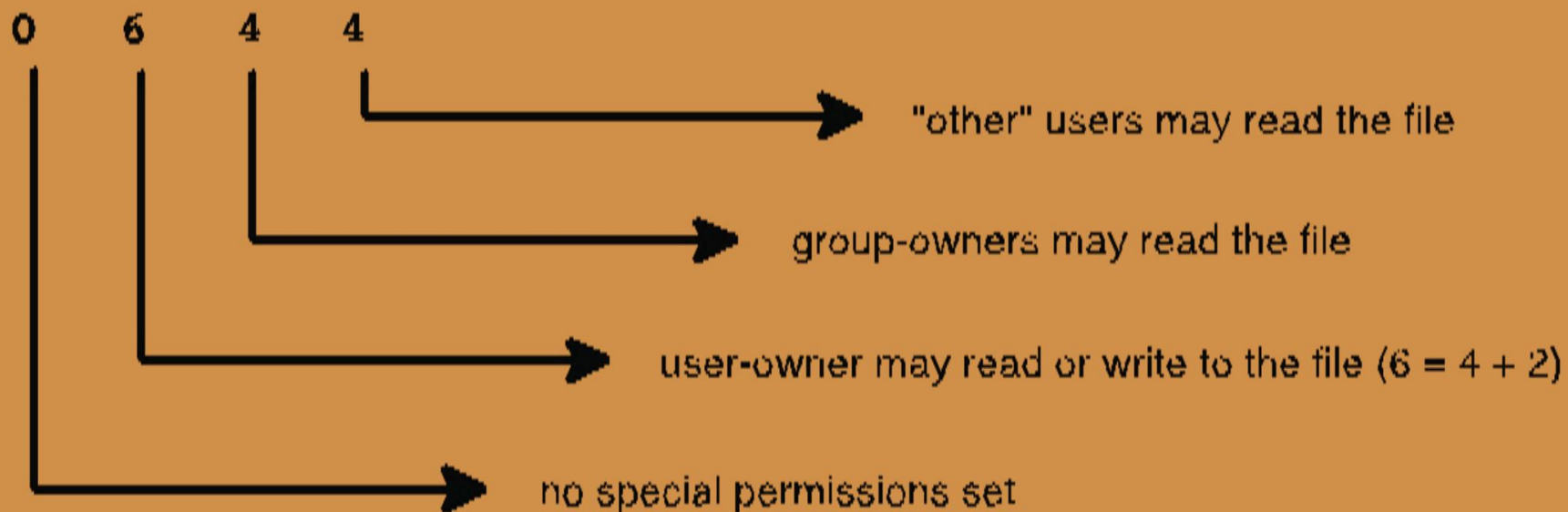


Figure 25.2 Permissions on mycoolfile

Espacio del núcleo versus Espacio de usuario

Espacio del núcleo

- Se refiere a la memoria utilizada por el kernel de Linux y sus módulos cargables.
 - o por ejemplo, controladores de dispositivos
- Debido a que el kernel aplica el DAC de Linux, es extremadamente importante aislar el espacio del kernel del espacio del usuario.
 - o Por este motivo, el espacio del núcleo nunca se intercambia con el disco duro.
 - o También es la razón por la que sólo el root puede cargar y descargar módulos del kernel.

Espacio de usuario

- Se refiere a la memoria utilizada por todos los demás procesos.


Vulnerabilidades de Linux

- Algunas vulnerabilidades comunes en Linux predeterminado

Las instalaciones (sin parches y sin seguridad) han sido:

- o Desbordamientos de búfer
- o Condiciones de carrera
- o Abuso de programas ejecutados “setuid root”
- o Denegación de servicio (DoS)
- o Vulnerabilidades de aplicaciones web
- o Ataques de rootkit

Abuso de programas Ejecutar “setuid root”



Un programa raíz setuid es un programa propiedad de la raíz con su bit setuid establecido: un programa que se ejecuta como raíz sin importar quién lo ejecute.


Si un programa root setuid puede ser explotado o abusado de alguna manera, entonces usuarios sin privilegios podrían usar ese programa para ejercer privilegios root no autorizados --- posiblemente abriendo un shell root (una sesión de línea de comandos ejecutándose con privilegios root)

La ejecución de setuid root es necesaria para los programas que deben ser ejecutados por usuarios sin privilegios pero que deben proporcionar a dichos usuarios acceso a funciones privilegiadas; por ejemplo, cambiar su contraseña, lo que requiere cambios en los archivos del sistema protegidos.

Un programa propiedad de root solo debe tener su bit setuid establecido si es absolutamente necesario

Aplicación web Vulnerabilidades


Esta es una categoría muy amplia de vulnerabilidades, muchas de las cuales caen en otras categorías de la lista.



Mientras que las aplicaciones web están escritas en lenguajes de scripting como PHP, Perl y Java pueden no ser tan propensos a desbordamientos de búfer clásicos, pero son propensos a abusos similares de manejo de entrada deficiente.



Hoy en día, pocas distribuciones de Linux se entregan con aplicaciones web “habilitadas de forma predeterminada”



Sin embargo, muchos usuarios instalan aplicaciones web con vulnerabilidades conocidas o escriben aplicaciones web personalizadas que tienen fallas que son fáciles de identificar y explotar.

Ataques de rootkit

Este ataque, que permite a un atacante ocultar sus huellas, generalmente ocurre después de un compromiso de la raíz.

Los rootkits comenzaron como colecciones de “reemplazos pirateados” para programas comunes. Comandos de UNIX que se comportaban como los comandos legítimos que reemplazaban, excepto que ocultaban los archivos, directorios y procesos de un atacante

Un rootkit de módulo de kernel cargable (LKM) cubre las huellas de los atacantes en el espacio del kernel, interceptando las llamadas del sistema relacionadas con los intentos de cualquier usuario de ver el intruso. recursos

Además de operar a un nivel más bajo y global, otra ventaja del rootkit LKM sobre los rootkits tradicionales es que las herramientas de verificación de integridad del sistema como Tripwire no generará alertas de los comandos del sistema que se reemplacen

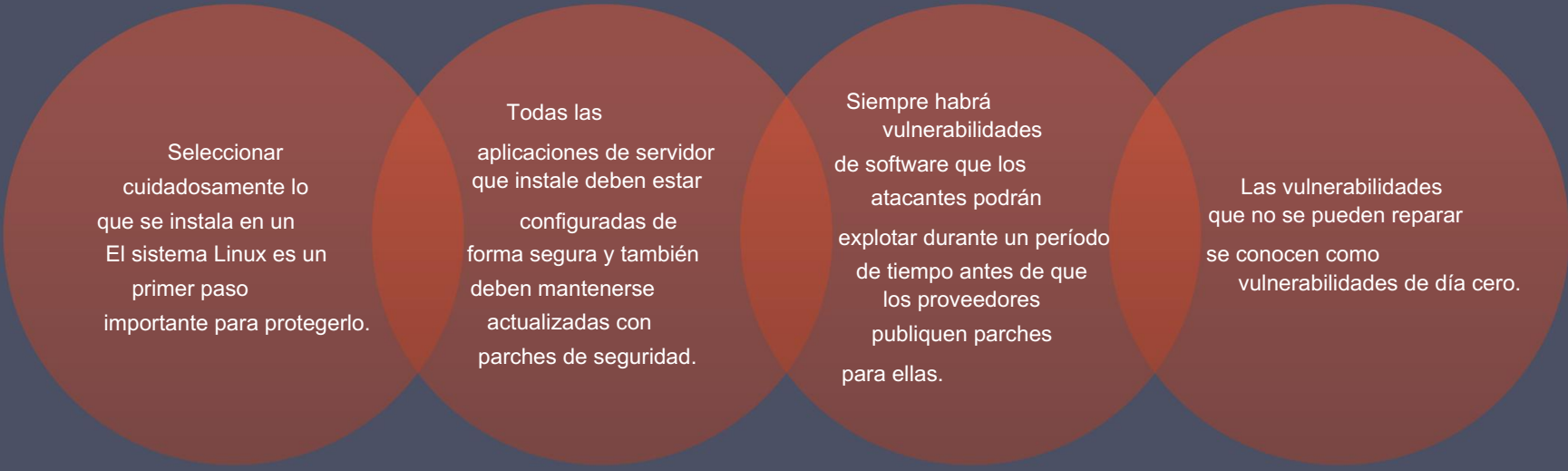
Muchos rootkits tradicionales y LKM se pueden detectar con el script chkrootkit

Instalación del sistema operativo: software

Selección y configuración inicial

- La seguridad del sistema Linux comienza en el momento de la instalación del sistema operativo.
- Aquí hay una lista de paquetes de software que debería rara vez, o nunca, se instalan en servidores protegidos, especialmente en servidores que dan a Internet:
 - o Sistema X Windows
 - o Servicios RPC
 - o Servicios R
 - el inetd
 - o Demonios SMIP
 - o Telnet y otros servicios de inicio de sesión en texto sin formato

Gestión de parches



Seleccionar cuidadosamente lo que se instala en un sistema Linux es un primer paso importante para protegerlo.

Todas las aplicaciones de servidor que instale deben estar configuradas de forma segura y también deben mantenerse actualizadas con parches de seguridad.

Siempre habrá vulnerabilidades de software que los atacantes podrán explotar durante un período de tiempo antes de que los proveedores publiquen parches para ellas.

Las vulnerabilidades que no se pueden reparar se conocen como vulnerabilidades de día cero.

Envoltorios de bibliotecas y

Envoltorios TCP

- Uno de los sistemas de control de acceso a red más maduros.
Los mecanismos en Linux son libwrappers
- En su forma original, el paquete de software TCP
Envoltorios, el demonio tcpd se utiliza como un proceso
contenedor para cada servicio iniciado por inetd
- Antes de permitir una conexión a cualquier servicio determinado
tcpd primero evalúa los controles de acceso
 - o Si la transacción coincide con alguna regla en hosts.allow, está permitida.
 - o Si no coincide ninguna regla en hosts.allow, se evalúa la transacción.
contra las reglas en hosts.deny

Envoltorios de bibliotecas y

Envoltorios TCP

- Los controles de acceso se basan en
 - o El nombre del servicio local al que se está conectando
 - o La dirección IP de origen o el nombre de host del cliente que intenta conexión
 - o El nombre de usuario del cliente que intenta la conexión
- La mejor manera de configurar los controles de acceso de TCP Wrappers es establecer una política de “denegar todo” en `hosts.deny`
- Los TCP Wrappers ya no se utilizan tan comúnmente como `libwrappers`

Las aplicaciones compatibles con `libwrapper` pueden usar los controles de acceso en `hosts.allow` y `hosts.deny` a través de llamadas al sistema proporcionadas por `libwrappers`

Usando iptables para Reglas del firewall local

- Los libwrappers y TCP Wrappers no son tan potentes como el mecanismo de firewall nativo del kernel de Linux, netfilter (iptables).
- iptables es muy útil cuando se ejecuta en un firewall con múltiples interfaces.
sistemas que protegen redes grandes como cuando se ejecutan en servidores comunes y sistemas de escritorio para protección local
- Casi todas las distribuciones de Linux ahora incluyen utilidades para generar automáticamente reglas de firewall locales



Software antivirus

Históricamente
Linux no ha sido
tan vulnerable a
los virus como
otros sistemas
operativos

La mayoría de
los
administradores de
sistemas Linux
tienden a
confiar en
mantenerse
actualizados
con parches de
seguridad
para
protegerse contra malware.

Históricamente,
los gusanos
han sido una
amenaza mucho
mayor para los
sistemas Linux
que los virus.

Los virus
suelen
abusar de los
privilegios de
cualquier usuario
que los ejecute
sin saberlo.

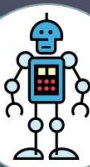
A medida que
la popularidad
de Linux continúa
creciendo, podemos
esperar que los
virus de Linux
se vuelvan
mucho más comunes.
común



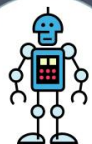
Gestión de usuarios

- Los principios rectores de la seguridad de las cuentas de usuario de Linux son:
 - o Tenga mucho cuidado al configurar permisos de archivos y directorios
 - o Utilice membresías de grupo para diferenciar entre diferentes roles en su sistema
 - o Tenga mucho cuidado al otorgar y utilizar privilegios de root.
- Revisión de comandos:
 - El comando `chmod` establece y cambia los permisos de los objetos que pertenecen a usuarios y grupos existentes
 - o `useradd`, `usermod` y `userdel` se utilizan para crear, modificar y eliminar cuentas de usuario
 - Los comandos `groupadd`, `groupmod` y `groupdel` se utilizan para crear, modificar y eliminar cuentas de grupo.

Caducidad de contraseñas



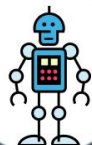
Duración máxima y mínima de las contraseñas de usuario



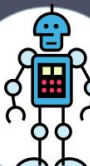
Establecido globalmente en los archivos `/etc/login.defs` y `/etc/default/useradd`



Las contraseñas deben tener una antigüedad mínima para evitar que los usuarios cambien rápidamente de contraseña en un intento de reutilizar contraseñas antiguas.



Si la edad máxima es demasiado larga, aumentarán las probabilidades de que las contraseñas se expongan antes de ser cambiadas; sin embargo, si es demasiado corta, los usuarios pueden frustrarse por tener que cambiar sus contraseñas con frecuencia, lo que lleva al mal uso de sus contraseñas.



Las cuentas de usuario inactivas deben deshabilitarse o eliminarse lo antes posible.

Delegación raíz: su and sudo

- El problema fundamental con Linux y UNIX

La seguridad es que los permisos y la autoridad en un sistema determinado se reducen a "la raíz puede hacer cualquier cosa, los usuarios no pueden hacer casi nada".

- su

- o Siempre que conozca la contraseña de root, puede usar el comando su para promocionarse a root desde cualquier usuario con el que haya iniciado sesión.

- sudo

- o Abreviatura de "superusuario hacer"

- o Paquete estándar en la mayoría de distribuciones de Linux

- o Permite a los usuarios ejecutar comandos específicos como root sin necesidad de hacerlo.

- Necesito saber la contraseña de root

Explotación florestal

Sólo puedo contarte
cosas malas que ya
sucedieron.

Los daemons de registro del
sistema reciben datos de registro
de una variedad de fuentes,
los ordenan por instalación
y gravedad, y luego
escriben los mensajes de registro
en archivos de registro.

En los sistemas Linux,
los registros del
sistema son manejados por el
Demonio Syslog de
Berkeley junto
con el demonio de registro
del núcleo o mediante Syslog-
NG

¿No es un?

control proactivo

Ayuda a garantizar que,
en caso de una violación o
falla del sistema, los
administradores del sistema
puedan identificar con mayor
rapidez y precisión lo
que sucedió.

Facilities	Facility Codes†	Priorities (in increasing order)	Priority Codes†	Actions
auth	4	none	n/a	/some/file(log to specified file)
auth-priv	10	debug	7	-/some/file (log to spec'd file but don't sync afterwards)
cron	9	info	6	/some/pipe (log to specified pipe)
daemon	3	notice	5	dev/some/tty_or_console (log to specified console)
kern	0	warning	4	@remote.hostname.or.IP (log to specified remote host)
lpr	6	err	3	username1, username2, etc (log to these users' screens)
mail	2	crit	2	* (log to all users' screens)
mark	n/a	alert	1	
news	7	emerg	0	
syslog	5	* ('any priority')	n/a	
user	1			
uucp	8			
local {0-7}	16-23	Usage of ! and = as prefixes with priorities		
* {'any facility'}	n/a	*.notice (no prefix)	=	"any event with priority of notice or higher"
		*!.notice	=	"no event with priority of notice or higher"
		*.=notice	=	"only events with priority of notice"
		*.!=notice	=	"no events with priority of notice"

†Numeric facility codes should not be used under Linux; they're here for reference only, as some other syslogd implementations (e.g., Cisco IOS) do use them

Figure 25.3 Syslogd Reference

Explotación florestal

- Syslog-NG o

- Creación del desarrollador húngaro Balazs Scheidler

- o Es preferible a syslogd porque (1) puede utilizar una variedad mucho más amplia de orígenes y destinos de datos de registro y (2) su motor de reglas es mucho más flexible que el archivo de configuración simple de syslogd, lo que le permite crear un conjunto de reglas mucho más sofisticado para evaluar y procesar datos de registro

- o Admite registro a través de TCP que puede cifrarse

- Tanto syslogd como Syslog-NG se instalan con configuraciones predeterminadas para lo que se registra y dónde.

- o Debe decidir qué combinación de registro local y remoto utilizar
llevar a cabo

- o Si los registros permanecen locales en el sistema que los genera, un atacante puede alterarlos.

- o Si algunos o todos los datos de registro se transmiten a través de la red a algún servidor central
Los registros de auditoría del servidor de registros se pueden conservar de manera más eficaz, pero los datos de registro también pueden quedar expuestos a espías de la red.

Otros sistemas de seguridad

Herramientas

Bastilla

Un sistema integral
de refuerzo
ng utilidad que
educa como tal
asegura

Cable trampa

Una utilidad
que mantiene
una base de
datos de
características
de archivos
cruciales del
sistema e informa
todos los cambios realizados en ellos.

Bufido

Un potente y gratuito
Intrusión
Detección
Sistema (IDS)
que detecta
común
ataques basados
en red

Neso

Un escáner
de
seguridad
modular que busca
común
Vulnerabilidades
del sistema y
de las aplicaciones

Corriendo como un Usuario/grupo sin privilegios

Una de las características de seguridad más importantes que puede tener un demonio es la capacidad de ejecutarse como un usuario o grupo sin privilegios.

Es posible que el proceso padre de un servicio se ejecute como root para vincularse a un puerto privilegiado y luego generar un nuevo proceso hijo que se ejecute como un usuario sin privilegios cada vez que se realiza una conexión entrante.

Lo ideal sería que los usuarios y grupos sin privilegios utilizados por un demonio de red determinado estuvieran dedicados a ese propósito.

Corriendo en una cárcel chroot

- La llamada al sistema chroot confina un proceso a un subconjunto de /
- Asigna un “/” virtual a algún otro directorio (por ejemplo, /srv/ftp/public)
- El directorio al que restringimos el demonio se llama cárcel chroot
- Para el demonio chrootado, todo lo que está en la cárcel chroot parece estar en realidad en /
- Las cosas en directorios fuera de la cárcel chroot no son visible o alcanzable en absoluto

Encriptación

El envío de credenciales de inicio de sesión o datos de aplicaciones a través de redes en texto sin formato los expone a ataques de escuchas clandestinas de la red.

La mayoría de las aplicaciones de red de Linux admiten el cifrado

El uso del cifrado a nivel de aplicación es la forma más eficaz de garantizar el cifrado de extremo a extremo de las transacciones de red.

Para una seguridad óptima, ya sea local o comercial
Se debe utilizar una autoridad de certificación (CA) para firmar todos los certificados de servidor, pero también se pueden utilizar certificados autofirmados.

Una computadora con MAC tiene una política de seguridad global a la que están sujetos todos los usuarios del sistema.

Un usuario que crea un archivo en un

Por lo general, es posible que el sistema MAC no se configure controles de acceso a ese archivo que son más débiles que los controles dictados por la política de seguridad del sistema.

Lo único para lo que se utiliza la cuenta de superusuario es para mantener la política de seguridad global.

Administración diaria del sistema
Se realiza

mediante cuentas que carecen de autoridad para cambiar la política de seguridad global.

Como resultado, es imposible comprometer todo el sistema atacando cualquier proceso.

SELinux

Potente implementación de controles de acceso obligatorios para Linux por parte de la NSA



No previene ataques de día cero, pero está diseñado específicamente para contener sus efectos.



Es una tecnología complicada y su configuración y resolución de problemas puede llevar mucho tiempo.



Incluso bajo SELinux los DAC de Linux todavía se aplican



Evalúa las acciones intentadas por los sujetos contra los objetos.



Los “sujetos” son siempre procesos



Las acciones se llaman “permisos”

Contextos de seguridad:

Usuarios, Roles, Dominios

- Cada sujeto y objeto individual controlado por SELinux está gobernado por un contexto de seguridad, cada uno de los cuales consta de un usuario, un rol y un dominio (también llamado tipo).
- Usuario
 - o Un usuario individual, ya sea humano o demonio.
 - o SELinux mantiene su propia lista de usuarios separada del sistema DAC de Linux
- Role
 - o Un rol puede ser asumido por cualquiera de varios usuarios preautorizados, cada uno de los cuales puede estar autorizado a asumir diferentes roles en diferentes momentos.
 - o En SELinux, un usuario solo puede asumir un rol a la vez y solo puede cambiar de rol si y cuando esté autorizado para hacerlo
- Dominio
 - o Una combinación de sujetos y objetos que pueden interactuar entre sí.
 - o En SELinux, dominio y tipo son sinónimos

Contextos de seguridad

- Aplicación de tipo (TE)
 - o El corazón de SELinux
 - o Modelo en el que cada proceso (sujeto) está asignado a un dominio en el que sólo se permiten ciertas operaciones
 - o Constituye la mayor parte de la implementación de SELinux en Fedora y Red Hat Enterprise Linux



Toma de decisiones en SELinux

Hay dos tipos de decisiones
SELinux debe hacer lo siguiente
respecto a sujetos, dominios y objetos:

Decisiones de acceso

Involucrar a los sujetos haciendo cosas
con objetos que ya existen o creando cosas
nuevas que permanecen en el
dominio esperado.

Son fáciles de entender

Decisiones de transición

Implican la invocación de procesos en dominios
diferentes a aquel en el que se ejecuta el
proceso en cuestión, o la creación de objetos de
tipos diferentes a sus directorios principales.

Son necesarios porque el mismo archivo o
recurso puede usarse en múltiples dominios/
tipos

Modelos de seguridad

- Acceso basado en roles

Control (RBAC)

- o Proporciona controles especialmente
Útil cuando se trata de usuarios humanos reales,
a diferencia de demonios y otros procesos
automatizados.
- o Es relativamente sencillo
- o En RBAC un usuario no puede
asumir más de un rol a la vez

- Seguridad multinivel (MLS)

- o Basado en el método Bell-LaPadula
Modelo (BLP)
- o Se aplica mediante el etiquetado del sistema
de archivos.



Novell AppArmor

- Implementación MAC de Novell para SuSE
- Hace que la tecnología MAC sea una opción viable para los administradores de sistemas que desean controles de seguridad sólidos pero no tienen el tiempo ni la paciencia para configurar y mantener SELinux.
- Está construido sobre los módulos de seguridad de Linux
- El objetivo es restringir el comportamiento de determinados aplicaciones de una manera muy granular pero específica
- Se basa en el supuesto de que el mayor vector de ataque en la mayoría de los sistemas son las vulnerabilidades de las aplicaciones.

Resumen

- Modelo de seguridad de Linux

- El DAC de Linux en

Profundidad: seguridad del sistema de archivos

- o Usuarios, grupos y permisos

- o Permisos de archivos simples

- Permisos de directorio

- La parte pegajosa

- setuid y setgid

- setgid y directorios

- Modos numéricos

- Espacio del kernel versus espacio del usuario



- Vulnerabilidades de Linux

- o Abuso de programas ejecutados setuid root

- o Vulnerabilidades de aplicaciones web

- o Ataques de rootkit

- Fortalecimiento del sistema Linux

- o Instalación del sistema operativo: selección de software y configuración inicial

- o Gestión de parches

- Nivel de red

controles de acceso

- o Software antivirus

- o Registro

- Seguridad de la aplicación

- Controles de acceso obligatorios