Taller de seguridad para sistemas GNU/Linux

Asegurando nuestros servidores al máximo

Carlos López
Igalia
@cl0p3z

Javier Álvarez
BritishTelecom
@joakkinen

Índice I

Índice II

Conceptos básicos

Vulnerabilidad

- punto del sistema susceptible de ser atacado. Debilidad del sistema.

Amenaza

- riesgo para el sistema con posibilidades de materializarse. Puede ser una persona, un programa ó suceso.

Contramedida

- técnica de protección del sistema.

Características básicas I

Disponibilidad

- mantener el funcionamiento correcto y recuperarse en caso de fallo.

Integridad

- asegurar que la información no se ha manipulado.

Confidencialidad

- impedir el acceso a la información a quien no tenga autorización

Características básicas II

Autenticidad

- asegurar la identidad del origen de la información.

Consistencia

- el sistema se comporta como debería con los usuarios autorizados.

Aislamiento

- controla el acceso al sistema.

Auditoría

- registrar qué acciones se han llevado a cabo en el sistema. Cuándo y qué.

Principios fundamentales I

Principio de menor privilegio.

- cualquier actor del sistema (usuario, administrador, programa, ...) debe tener los privilegios necesarios para realizar su tarea y ninguno más.

Principio del eslabón más débil.

- la fortaleza un sistema de seguridad la marca la fortaleza de su eslabón más débil.

Principio del punto de control centralizado.

- un único punto de acceso en el que concentrar las medidas de seguridad.

Principios fundamentales II

Seguridad en caso de fallo.

- en caso de que el sistema de seguridad falle, el sistema debe permanecer en un estado seguro.

Participación universal.

- cualquier mecanismo de seguridad es vulnerable con la ayuda de un usuario autorizado.

Simplicidad.

- los sistemas complejos ocultan errores con mayor facilidad.

2. Protección física

2.1. Arranque.

2.2. Sistema de ficheros

2.3. Cifrado

Protección física

- Restringir el acceso físico:
  - Control de acceso.
  - Candados / alarmas.
  - Electricidad redundante.
  - Etc...

- Opciones de BIOS
  - Cargar directamente desde el disco principal
  - Protegerlo con contraseña

Arranque (grub)

/etc/default/grub:

```
GRUB_TIMEOUT=1
GRUB_DISABLE_RECOVERY="true”
```

/etc/grub.d/01_security: (grub-mkpasswd-pbkdf2)

```
#!/bin/sh
cat << EOF
set superusers="root"
password_pbkdf2 root grub.pbkdf2.sha512.10000.
F2A7830AC4F8BB6C99127BCD2C8D82B02F98697E1DB5433C70F8E3F5D562EC96B8E79
E6F28CB072238A0E391D3CB3A5B5E718790ABDA1E10CA0D92FF5668929A.
727939BEEA853745DD53F0D8170B1B137B0A49E0126C71349F9683878062C2AE9BA94
D9BA860A10D55859F8DA827134CFC0C7C2FB0770EA44FC40CB1E5370AFC
EOF
```

https://www.gnu.org/software/grub/manual/html_node/Security.html

Arranque (otros parámetros)

- Tecla “petición de sistema” o (Magic) SysRQ
  - Desactivarla con sysctl:
    - `echo 'kernel.sysrq = 0' > /etc/sysctl.d/sysrq.conf`

- Teclas “Ctrl+Alt+Del”
  - sysvinit:
    - `vi /etc/inittab` #ctrlaltdel
  - systemd:
    - `ln -s /dev/null /etc/systemd/system/ctrl-alt-del.target`
  - upstart:
    - `vi /etc/init/control-alt-delete.conf`

- Acceso de root a las consolas físicas
  - Desactivarlo en `/etc/securetty`

Práctica

- Proteger el arranque de GRUB con contraseña
- Desactivar las teclas ctrl+alt+supr

Encriptación.

- dm-crypt: Encriptación transparente a nivel de kernel.
  - Muy rápido (sobretodo si usas cifrado AES y tu procesador soporta AES_NI)

- Encriptar todo el sistema:
  - /dev/sda1 ~[100-200]Mb [ext4 /boot]
  - /dev/sda2 [dm-crypt]
    - LVM
      - vg-root
      - vg-swap
      - vg-tmp
      - …
  - El instalador de Debian (modo texto) permite crear este tipo de layout.

- Encriptar sólo ciertas particiones
  - Encriptar swap.

Encriptación (acceso remoto).

- Encriptar todo el sistema: Corte de luz. ¿Como introduzco la clave en remoto?
  - Intervención manual:
    - KVM / IPMI
    - Mini servidor ssh (dropbear) en initramfs
      - http://blog.neutrino.es/es/2011/unlocking-a-luks-encrypted-root-partition-remotely-via-ssh/
  - Arranque automático:
    - Mandos
      - https://wiki.recompile.se/wiki/Mandos
    - keyscript

Encriptación (crypttab).

- /etc/crypttab

```
#<target name>    <source device>            <key file>  <options>
sda2_crypt UUID=fab1dca3-1283-434g-9122-f6aacc7491bf none luks,discard
sdb2_crypt UUID=1b65fabc-1816-f135-af6e-bf3dd7ba4431 /root/.luks/sdb2_crypt.pwd luks
```

  - options:
    - keyscript=/path/script.sh: Imprime la clave en stdout.
    - noauto
    - swap = Ejecuta mkswap en el dispositivo.
    - initramfs
    - …. man crypttab

Particionar el disco

- ¿Porqué particionar?
  - Para restringir el espacio
    - Cambiar el tamaño sin reparticionar: LVM
  - Para usar un FS relativo a su función.
  - Por las opciones de montaje
    - **noexec**: no permite la ejecución binarios.
    - **nosuid**: el bit suid no toma efecto.
    - ro: sólo lectura.
    - **nodev**: no interpreta archivos de bloques o de caracteres.
    - (no)atime: (no)actualiza la hora de acceso.
    - owner: el usuario debe ser dueño del dispositivo (implica nosuid y nodev).
    - group: el usuario debe pertenecer al grupo dueño del dispositivo (implica nosuid y nodev).
    - default == rw, suid, dev, exec, auto, nouser, async
    - sync: I/O sincrónico
    - **acl**: Activar las listas de control de acceso POSIX.

Opciones de montaje

- [no]exec: No se deben ejecutar comandos en el /tmp. Posiblemente /usr, /bin, /sbin, sea los únicos lugares.
- /home y /tmp: son territorio de usuario y por lo tanto deben ir aparte.
- /var/log: puede crecer demasiado y afectar el funcionamiento. Debería ir en una partición independiente.
- ro: por su criticidad y estabilidad, /usr, /bin, /sbin, /etc, /boot y /lib pueden ser de solo lectura.
- nodev: a excepción de /dev, no se requiere interpretar dispositivos
- nosuid: a excepción de /usr y /bin el bit suid debe ser ignorado

Cifrando del sistema de ficheros

Ventajas:

- protección extra de los datos.
- robo ó pérdida

Sistemas objetivo:

- sistemas con acceso físico no restringido ó poco protegido (portátiles, móviles, pymes, ...)

Práctica I

Disco secundario /dev/sdb en la máquina virtual con CentOS.

Paso 1: preparar el disco

fdisk - crear la partición (sdb1)

Paso 2: formato y clave

cryptsetup luksFormat /dev/sdb1

Práctica II

Paso 3: desbloqueo del volumen

```
cryptsetup luksOpen /dev/sdb1 [nombre]
```

(/dev/mapper/[nombre])

Paso 4: formato

```
mkfs -t ext4 /dev/mapper/[nombre]
```

Práctica III

Paso 5: montaje y uso

```
mkdir /[nombre]
mount /dev/mapper/[nombre] /[nombre]
```

Paso 6:

```
umount /[nombre]
cryptsetup luksClose [nombre]
```

Práctica IV

Configurar el montaje automático de la partición cifrada

Paso 1: /etc/crypttab

```
[nombre]   /dev/sdb1  /root/passwd
…
```

Paso 2:/etc/fstab

```
…
/dev/mapper/[nombre]  /[nombre]  ext4   defaults   1 2
```

Práctica V

Paso 3: crear archivo de clave

```
echo -n ‘clave’ > /root/passwd
chmod 600 /root/passwd
```

Paso 4: registrar la clave

```
cryptsetup luksAddKey /dev/sdb1  /root/passwd
```

3. Protección de la red

3.1. Firewall

3.2. Monitorización de la red

3.3. Detección de intrusos

Firewall

- Todos los servidores y máquinas en general, deben tener un firewall local. Esto ayuda a mitigar intrusiones y ataques.
  - Error: solo configurar el firewall en el router principal.
- Políticas:
  - Por defecto se prohíbe todas las conexiones entrantes.
    - Se permiten una por una los tipos de conexión entrante
  - Por defecto se prohíbe todas las conexiones de enlace (forwarding)
    - Se permite uno por uno los tipos legítimos de forwarding/enrutamiento.
  - Se pueden denegar las conexiones salientes.
    - Ej: Un servidor de base de datos no tiene motivo para acceder a internet más allá de acceder a los repositorios de paquetes.

```
iptables -P OUTPUT DROP
iptables -A OUTPUT -p tcp -d 192.168.0.0/16 --sport mysql -j ACCEPT
iptables -A OUTPUT -p tcp -d ftp.es.debian.org --dport http -j ACCEPT
```

Firewall

- iptables
  - Parte integral del framework Netfilter del Kernel
  - Interceptar y manipular paquetes de red
    - Firewall
    - Traducción de direcciones (NAT)
    - Registro (LOG)
    - Etc..

- ip6tables
  - IPv6 (!)

iptables

- Multiples frontends
  - Configuring iptables manually is challenging for the uninitiated.

    Fortunately, there are many configuration tools (wizards) available to assist https://wiki.debian.org/Firewalls

- iptables
  - Shell script
    - /etc/init.d/firewall
    - /etc/network/if-up.d/firewall
  - iptables-save
    - Guarda las reglas actuales en un fichero
  - iptables-restore
    - Restaura las reglas actuales

iptables (Practica)

- Configurar el firewall en el servidor:
  - Para IPv4 denegar todo el tráfico entrante, excepto en:
    - Interfáz de loopback.
    - Conexiones ya establecidas.
    - Puerto SSH (tcp) solo permitir desde la intranet.
    - Puerto http y https (tcp) permitir desde cualquier IP.
    - Permitir ping (ICMP)
    - Logear todo el trafico entrante que vaya a ser denegado
  - Repetir para IPv6?

iptables

- Fun with iptables:
  - iptables -I INPUT -m u32 --u32 52=0x18030000:0x1803FFFF -j LOG --log-prefix 'HEARTBLEED '
  - iptables -I INPUT -m u32 --u32 52=0x18030000:0x1803FFFF -j DROP
  - ip6tables -I INPUT -m u32 --u32 52=0x18030000:0x1803FFFF -j LOG --log-prefix 'HEARTBLEED '
  - ip6tables -I INPUT -m u32 --u32 52=0x18030000:0x1803FFFF -j DROP

  - iptables -A INPUT -m string --algo bm --hex-string '|28 29 20 7B|' -j --log-prefix ‘SHELLSHOCK '
  - iptables -A INPUT -m string --algo bm --hex-string '|28 29 20 7B|' -j DROP
  - ip6tables -A INPUT -m string --algo bm --hex-string '|28 29 20 7B|' -j --log-prefix ‘SHELLSHOCK '
  - ip6tables -A INPUT -m string --algo bm --hex-string '|28 29 20 7B|' -j DROP

firewall: parametros del kernel

/etc/sysctl.d/firewall.conf

- Deshabilitar IPv6 ?

  `net.ipv6.conf.all.disable_ipv6=1`

- No responder ICMP echos broadcasts

  `net.ipv4.icmp_echo_ignore_broadcasts=1`

- Ignorar respuestas ICMP no pedidas

  `net.ipv4.icmp_ignore_bogus_error_responses=1`

- Ignorar ICMP redirects (a menos que haya 2 gateways)

  `net.ipv4.conf.all.accept_redirects=0`

  `net.ipv4.conf.default.accept_redirects=0`

- No aceptamos paquetes con "source route"

  `net.ipv4.conf.all.accept_source_route=0`

  `net.ipv4.conf.default.accept_source_route=0`

knockd

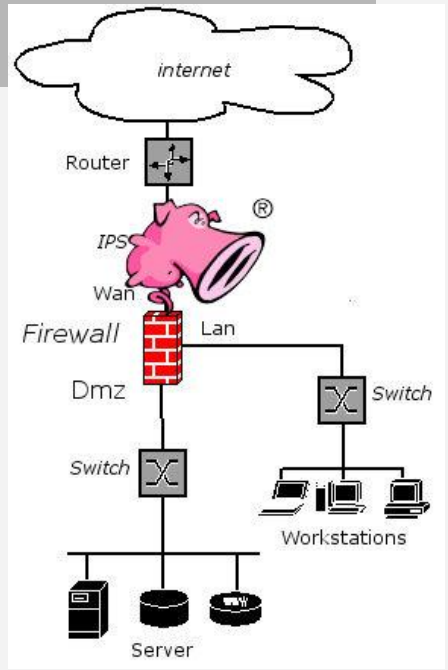
- Port knocking (Tocar puertos) es un método discreto de abrir puertos que, por default, el firewall mantiene cerrado. Funciona requiriendo intentos de conexión a una serie de puertos predefinidos cerrados. Cuando la secuencia correcta de "toquidos" a puertos (intentos de coneccion) es recibida, el firewall abre entonces cierto(s) puerto(s).

- El beneficio es que, en un escaneo de puertos normal, parecería que el servicio del puerto simplemente no esta disponible.

Práctica (knockd)

- Configurar knockd
  - Abrir el puerto SSH a la IP que intente conectar de forma secuencial con:

4. Detección de intrusos



- Network IDS:
  - SNORT
    - Es un sistema de detección de intrusiones de red («Network Intrusion Detection System»). Su función es escuchar la red y tratar de detectar intentos de infiltración y/o actos hostiles (inclusive ataques de denegación de servicio). Todos estos eventos son registrados y diariamente se envía un email al administrador con un resumen de las últimas 24 horas.
    - **Internet <-> Firewall <----------> LAN .**
    - **^----------^-------- SNORT ----^ .**
      - Antes del firewall / Después del firewall
    - https://www.snort.org/documents
    - https://www.debian-administration.org/article/318
    - http://debian-handbook.info/browse/es-ES/stable/sect.supervision.html
    - apt-get install snort

Detección de intrusos

- Monitorización de archivos.
  - AIDE
    - La herramienta AIDE (entorno avanzado de detección de intrusión: «Advanced Intrusion Detection Environment») permite comprobar la integridad de los archivos y detectar cualquier cambio frente a una imagen guardada previamente del sistema válido. Se almacena esta imagen como una base de datos (/var/lib/aide/aide.db) que contiene la información relevante de todos los archivos del sistema (huella digital, permisos, marcas temporales, etc.).
  - Verificar integridad de los paquetes instalados:
    - debsums
    - rpm -Va

Detección de intrusos

- Monitorización de usuarios.
  - snoopy
    - Wrapper de execve(). Loggea via syslog.
    - Dos modos de funcionamiento:
      - /etc/ld.so.preload
        - Registra toda la actividad de los usuarios en el sistema.
        - Puede ocasionar perdida de rendimiento o llenar el disco debido a la cantidad de información loggeada.
      - export LD_PRELOAD=/lib/snoopy.so
        - Registra la actividad de determinadas aplicaciones
    - Util: rsyslog
    - apt-get install snoopy
    - tail -f /var/log/auth.log

Detección de intrusos

- Monitorización de usuarios.
  - GNU Accounting Utilities
    - “last” con esteroides.
      - Loggea un resumen de los comandos ejecutados por los usuarios.
    - apt-get install acct
    - sa --print-users
    - lastcomm --user gsickminds
    - lastcomm rm
  - En tiempo real:
    - finger
    - whowatch
    - htop / top -u gsickminds
    - netstat -an**p**[tu]

Practica

- Instalar y configurar AIDE
  - /etc/aide/aide.conf
  - aideinit
  - cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db
  - aide -c /etc/aide/aide.conf --check
  - cron → mail
- Comprobar los paquetes con debsums
- Probar snoopy

5. Control de acceso I

5.1. Control de acceso discrecional (DAC)

    5.1.1. Elevación de permisos (sudo)

    5.1.2. Permisos, permisos especiales y ACLs

Evitando a root

- Como buena práctica, se deben evitar los usuario impersonales.
  - root es un usuario impersonal.
  - Luego, se debe evitar el uso de root.
- Herramientas disponibles:
  - sudo
  - acceso a las terminales locales
    - access.conf
    - securetty
  - acceso remoto
  - grupos privilegiados
    - adm (acceso a logs)
    - disk (acceso raw a dispositivos de bloques (discos))
    - sudo (acceso sudo)
    - … https://www.debian.org/doc/manuals/securing-debian-howto/ch12.en.html
  - el bit SUID
  - capabilities

sudo

- Por defecto los usuarios del grupo sudo

  Privilegio del grupo sudo

  %sudo ALL=(ALL:ALL) ALL

- **/etc/sudoers**
  - visudo
  - EDITOR=nano visudo
  - **[user|%group] hosts** = [(**runas user:group**)] [NOPASSWD: | PASSWD:] [NOEXEC:] **cmd**

- Evitar escapes exec
  - NOEXEC: /usr/bin/vim,/bin/more,/bin/less

sudo (práctica)

- Permitir al usuario developer editar los ficheros de configuración de apache.

sudo (práctica)

- `/etc/sudoers:`
  - `developer ALL=(ALL:ALL) /usr/bin/vi /etc/apache2/*`
    - `?`

sudo (práctica)

- `/etc/sudoers:`
  - `developer ALL=(ALL:ALL) /usr/bin/vi /etc/apache2/*`
    - Problema de seguridad (shell escape)
  - `developer ALL=(ALL:ALL)NOEXEC: /usr/bin/vi /etc/apache2/*`

sudo (práctica)

- `/etc/sudoers:`
  - `developer ALL=(ALL:ALL) /usr/bin/vi /etc/apache2/*`
    - Problema de seguridad (shell escape)
  - `developer ALL=(ALL:ALL)NOEXEC: /usr/bin/vi /etc/apache2/*`
    - Y si a developer no le gusta vi?

sudo (práctica)

- /etc/sudoers:
  - developer ALL=(ALL:ALL) /usr/bin/vi /etc/apache2/*
    - Problema de seguridad (shell escape)
  - developer ALL=(ALL:ALL)NOEXEC: /usr/bin/vi /etc/apache2/*
    - Y si a developer no le gusta vi?
- developer ALL=(ALL:ALL) sudoedit /etc/apache2/*, sudoedit /etc/apache2/*/*
  - export EDITOR=myeditor
  - Se ejecuta con permisos del usuario.
    - El fichero que se edita reside en /tmp.

capabilities

- **Realmente necesitas ser root?**
  - **# ls -l /bin/ping**

    -rwsr-xr-x 1 root root 31104 Apr 12 2011 /bin/ping
  - **Existe una excesiva disparidad entre los privilegios de un usuario root y uno no-root**
  - **SUIDes un privilegio “full power” temporal**
- **Capabilities es una forma de dividir el poder de root**

```
$ cp /bin/ping .
$ ./ping google.es
ping: icmp open socket: Operation not permitted
$ strace ./ping google.com 2>&1 |grep EPERM
socket(PF_INET, SOCK_RAW, IPPROTO_ICMP) = -1 EPERM (Operation not permitted)
$ sudo setcap cap_net_raw=ep ./ping
$ ./ping google.com
PING google.com (178.60.128.59) 56(84) bytes of data.
64 bytes from cache.google.com (178.60.128.59): icmp_req=1 ttl=63 time=14.8 ms
$ getcap ./ping
ping = cap_net_raw+ep
```

capabilities - ejercicio

- Que capabilities existen?
  - # man capabilities
- Ejercicio: Reemplazar el uso del bit setuid con capabilities

```
$ find {,/usr}/{,s}bin -user root -perm -4000 -exec ls -l '{}' \;
-rwsr-xr-x 1 root root 35252 Apr 12  2011 /bin/ping6
-rwsr-xr-x 1 root root 31104 Apr 12  2011 /bin/ping
-rwsr-xr-x 1 root root 88744 Dec  9  2012 /bin/mount
-rwsr-xr-x 1 root root 35200 May 25  2012 /bin/su
-rwsr-xr-x 1 root root 67704 Dec  9  2012 /bin/umount
-rwsr-xr-x 1 root root 66196 May 25  2012 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 45396 May 25  2012 /usr/bin/passwd
-rwsr-xr-x 2 root root 119172 Mar  1  2013 /usr/bin/sudoedit
-rwsr-xr-x 1 root root 35892 May 25  2012 /usr/bin/chsh
-rwsr-xr-x 2 root root 119172 Mar  1  2013 /usr/bin/sudo
-rwsr-xr-x 1 root root 44564 May 25  2012 /usr/bin/chfn
-rwsr-xr-x 1 root root 30880 May 25  2012 /usr/bin/newgrp
```

capabilities - ejercicio

- Que capabilities existen?
  - # man capabilities
- Ejercicio: Reemplazar el uso del bit setuid con capabilities

```
ping - CAP_NET_RAW (13)
chsh - CAP_CHOWN (0), CAP_DAC_READ_SEARCH (2), CAP_FSETID (4), CAP_SETUID
   (7)
chfn - CAP_CHOWN (0), CAP_DAC_READ_SEARCH (2), CAP_FSETID (4), CAP_SETUID
   (7)
passwd - CAP_CHOWN (0), CAP_DAC_OVERRIDE (1), CAP_FOWNER (3)
mount - CAP_DAC_OVERRIDE (1), CAP_SYS_ADMIN (21)
umount - CAP_DAC_OVERRIDE (1), CAP_SYS_ADMIN (21)
```

Capabilities – Desventajas

- ¿Por qué no viene out-of-the-box?
  - Requiere soporte para atributos extendidos (NFS no los soporta) Hay muchas aplicaciones que ignoran xattr
    - mv
    - cp (cp -a)
    - tar
    - rsync
- Faltan tools para administrarlos
- Es algo relativamente nuevo
- Existe debate si ciertas capabilities no son root-equivalent: http://forums.grsecurity.net/viewtopic.php?f=7&t=2522

Mi primer backdoor usando caps

```
#include <unistd.h>
int main(void)
{
        if (
               setuid(0) < 0 ||
               execl("/bin/bash", "bash", NULL) < 0
           )
                    perror("error..");
        return 1;
}
```

- `$ gcc backdoor.c  -o backdoor`
- `$ sudo setcap "cap_setuid=ep" ./backdoor`

ACL

- Son los permisos a nivel de usuario / grupo / otros suficientes?
  - ACL nos permite especificar permisos especificos para un usuario o grupo determinado
- No todos los sistemas de ficheros soportan ACL

```
$ # grep POSIX_ACL /boot/config-$(uname -r)
CONFIG_EXT2_FS_POSIX_ACL=y
CONFIG_EXT3_FS_POSIX_ACL=y
CONFIG_EXT4_FS_POSIX_ACL=y
CONFIG_REISERFS_FS_POSIX_ACL=y
CONFIG_JFS_POSIX_ACL=y
CONFIG_XFS_POSIX_ACL=y
CONFIG_BTRFS_FS_POSIX_ACL=y
CONFIG_FS_POSIX_ACL=y
CONFIG_TMPFS_POSIX_ACL=y
CONFIG_JFFS2_FS_POSIX_ACL=y
CONFIG_9P_FS_POSIX_ACL=y
```

- mount -o remount,acl /
- getfacl / setfacl

ACL (práctica)

```
# ls -l /etc/shadow
-rw-r----- 1 root shadow 884 Oct 28 14:48 /etc/shadow
# getfacl /etc/shadow
    # file: etc/shadow
    # owner: root
    # group: shadow
    user::rw-
    group::r--
    other::---
# setfacl -m u:developer:r /etc/shadow
# getfacl /etc/shadow
    # file: etc/shadow
    # owner: root
    # group: shadow
    user::rw-
    user:developer:r--
    group::r--
    mask::r--
    other::---
# ls -l /etc/shadow
    -rw-r-----+ 1 root shadow 884 Oct 28 14:48 /etc/shadow
```

- **Modificar permisos:**
  - **$ setfacl -m rules files**
    - **rules => u:uid:perms**
    - **rules => g:gid:perms**
- **Borrar permisos:**
  - **$ setfacl -x rules**
- **$ man setfacl**

- **Ejercicio: Permitir al usuario developer editar los ficheros de configuracion de apache (/etc/apache2) usando ACLs**

Eso es todo?

- /home/gsickminds/borrame.txt
  - Puedes borrarlo?
    - Sudo?
    - ACL?
    - SeLinux?
    - ????

Extended file attributes

- lsattr /home/gsickminds/borrame.txt
- man chattr
  - chflags en BSD/MacOS
- Feature originalmente solo disponible en ext2/ext3/ext4.
  - Ha sido portada a otros sistemas de ficheros (XFS,BRTFS,...)
  - No todos los sistemas de ficheros soportan todos los flags
- chattr +i
  - Fichero inmutable (no se puede borrar/modificar/..)
- chattr +a
  - Fichero append-only (solo se puede añadir contenido, no modificar el actual)
    - Útil: .bash_history
- chattr +S
  - Cualquier escritura en este fichero se hará de forma síncrona.
- ....

5. Control de acceso II

5.3. PAM

5.3. Control de acceso mandatorio (MAC)

- 4.3.1. Linux LSMs: Appamor, SELinux, SMACK
- 4.3.2. GrSecurity y PaX

PAM

- Pluggable Authentication Modules (PAM)
  - ¿Que es PAM? http://bit.do/PAM-es
    - Mecanismo flexible para la autenticación de usuario.
  - Ejemplos
    - pam_cracklib: Deniega el acceso si el password fue encontrado en un diccionario.
    - pam_time: Deniega el acceso a determinadas horas.
    - pam_google_authenticator: Integra el sistema de autenticación 2-factor de google.

PAM (Ejercicio)

- Configurar pam_cracklib
- /etc/pam.d/common-password en Debian (o /etc/pam.d/system-auth en CentOS)

```
password    requisite     pam_cracklib.so retry=3 minlen=12 difok=6 lcredit=1 ucredit=1 dcredit=2
   ocredit=1
```

  - 3 oportunidades para elegir un password fuerte.
  - con 12 caracteres minimo
  - con 6 caracteres diferentes del anterior password
  - Con ‘lcredit’ minúsculas, ‘ucredit’ mayúsculas, ‘dcredit’ dígitos y ‘ocredit’ otros signos.

Linux LSMs

- Linux Security Modules (LSM):
  - Framework en el Kernel de Linux.
    - LSM inserts "hooks" (upcalls to the module) at every point in the kernel where a user-level system call is about to result in access to an important internal kernel object such as inodes and task control blocks.
- LSM disponibles en el kernel oficial:
  - SeLinux
    - CentOS / RHEL
  - AppArmor
    - Ubuntu
  - SMACK
    - Tizen
  - TOMOYO
- Solo puede activarse un LSM a la vez.
  - /proc/cmdline => “apparmor=1 security=apparmor”

Grsecurity + PaX + RBAC

- Grsecurity:
  - Parche “extraoficial”.
    - Los desarrolladores del kernel no están interesados en integrarlo.
    - El desarrollador de Grsecurity (Spender) tampoco está interesado :)
      - “Paranoid” mode.
      - Seguridad ante todo (rendimiento no importa)
  - PaX
    - Mínimo privilegio páginas de memoria.
      - Páginas no ejecutables.
      - ASLR
      - …
  - RBAC
    - Alternativa a los diferentes LSM
      - No utiliza el framework LSM de Linux
  - Protege contra la mayoría de los exploits / buffer overflows.
  - https://grsecurity.net/features.php

6. Protección de la capa de aplicación

Chroot

Containers

Limitar recursos (inicios de sesión, cuotas)

Chroot

- Chroot (Change root):
  - Sirven para aislar a un proceso en un directorio.
  - No son seguros si el proceso tiene permisos de super usuario.
    - Escapar de un chroot es posible.
    - Puede controlar (kill) a procesos fuera del chroot.
    - ….

- Container = Chroot con esteroides.
  - + namespaces
    - PID namespaces
    - Network namespaces
    - User namespaces
    - Filesystem namespaces (chroot)
  - + cgroups
    - Limitar y controlar el uso de recursos (CPU/memoria/disco..) de un conjunto de procesos.

Container

- Diversas tecnologías:
  - Upstream kernel:
    - Docker
    - LXC
    - Seguros solo en combinación con algun LSM (Apparmor/SeLinux)
  - Third party kernel patches:
    - OpenVZ
      - El más avanzado hasta la fecha en cuanto a features.
    - Vserver
      - Grsecurity+vserver patch
- Are Linux containers secure?
  - https://lwn.net/Articles/617842

Usuarios y accesos.

- Bomba fork
  - shell:
    - **fork(){ fork|fork& };fork**
  - python:
    - while True: os.fork()
  - ....

Practica:

- Limitar los recursos del usuario para prevenir una bomba fork.
- Limitar acceso mediante limits.conf

Add the following line to /etc/pam.d/common-auth:

```
auth required    pam_access.so
```

- Limitando los recursos
  - /etc/security/limits.conf

```
+:root:192.168.200.1 192.168.200.4 192.168.200.9
+:gsickminds:ALL
-:ALL:ALL
```

  - <domain> <type> <item> <value>
- memlock, nproc, maxlogins, priority, nice, etc...

Limitando recursos y accesos

- Eliminar los shells interactivos a quienes no lo necesitan
  - chsh -s /bin/false daemon www-data …
- Buscar archivos con bit SUID/SGID
  - find / -path /proc -prune -o -type f -perm +6000 -ls
  - chmod ug-s <archivo>
- Modificar los plazos de las contraseñas de un usuario
  - chage <usuario>
- Control de acceso
  - /etc/security/access.conf
  - permission(+,-):users:origins
- Permisos para ejecutar tareas diferidas
  - /etc/cron.allow /etc/at.allow

7. SELinux

7.1. Conceptos generales

7.2. Modos de funcionamiento

7.3. Manejo de contextos

7.4. Uso de booleanos

7.5. Monitorizar violaciones

7.6. Metodología de trabajo

Introducción I

Capa de seguridad adicional para el sistema.

- implementación de MAC en Linux (control de acceso obligatorio)
- control de aplicaciones y usuarios -> archivos, dispositivos, red y comunicaciones entre procesos

Módulo para el kernel + herramientas

Desarrollado inicialmente por la NSA

En 2001 se publica el código

Introducción II

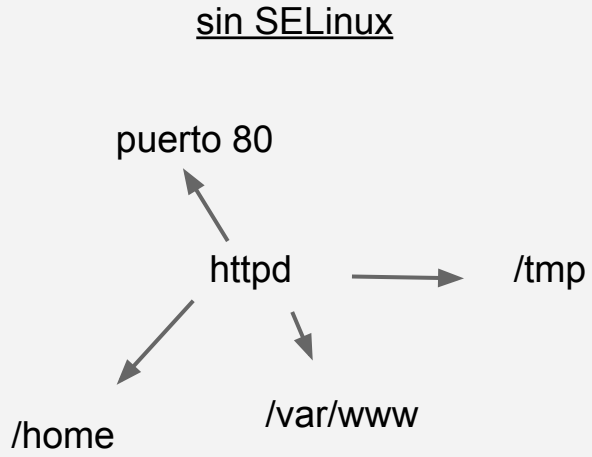
Incluido en el kernel 2.6.0-test3 2003

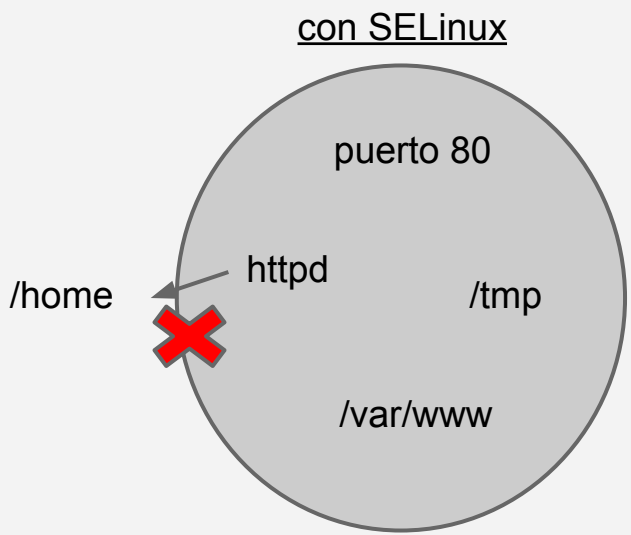
Disponible para todas las distribuciones

Usado en redhat (RHEL) por defecto en 4+

Introducción III

Construye una capa de protección alrededor la aplicación





Pregunta

Además de servidores, ¿en qué entorno puede ser interesante?

pista: aplicaciones de distintos fabricantes/desarrolladores, permisos por aplicación,

Pregunta

Efectivamente,

¡Móviles!!

SEAndroid (basado en SELinux). desde android 4.3.

Proporciona una protección extra entre aplicaciones, y entre las aplicaciones y el sistema.

SOFTWARE

Android Lollipop se toma más en serio la seguridad

ABC TECNOLOGÍA / MADRID | Día 29/10/2014 - 09.11h

▸ La versión 5.0 del sistema operativo de Google incorpora más funciones para proteger al usuario y sus datos

ACB - tecnología hoy

“La tercera clave de seguridad es el sistema Security Enhanced Linux (SELinux). Esto va de cara a los desarrolladores de las aplicaciones, quienes a partir de ahora deberán incorporar el modo SELinux en sus apps para todos los dispositivo. Ludwig apunta que desde que se introdujera SELinux el año pasado, se han evitado múltiples vulnerabilidades. “

Ventajas

- más potente, grano fino.

(leer, escribir, ejecutar -> unlink, append only, move a file)

- políticas de seguridad al margen de la configuración app y usuarios.

- protección extra en caso del mal comportamiento de la aplicación.

Práctica

Instalación (paquetes)

- policycoreutils
- policycoreutils-python
- setroubleshoot-server
- selinux-policy-doc

Modos de funcionamiento I

Modos:

desactivado

activado

- permissive
- enforcing

Comprobar:

sestatus

Modos de funcionamiento II

Cambio en caliente:

- getenforce
- setenforce 0|1

Cambio en frío:

/etc/sysconfig/selinux

Primer vistazo

ls -Z

```
[root@localhost gsick]# ls -lZ
drwxr-xr-x. gsick gsick unconfined_u:object_r:user_home_t:s0 Descargas
drwxr-xr-x. gsick gsick unconfined_u:object_r:user_home_t:s0 Documentos
drwxr-xr-x. gsick gsick unconfined_u:object_r:user_home_t:s0 Escritorio
drwxr-xr-x. gsick gsick unconfined_u:object_r:user_home_t:s0 Imágenes
drwxr-xr-x. gsick gsick unconfined_u:object_r:user_home_t:s0 Música
```

contexto SELinux: usuario:recurso:tipo:nivel

Segundo vistazo

```
ps -Z

unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 1674 pts/0 00:00:00 bash
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 1692 ? 00:05:55 firefox
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 1776 pts/0 00:00:00 su
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 1782 pts/0 00:00:00 bash
```

Ayuda

Ayuda - paginas man especiales

man -k selinux

(makewhatis)

Una página por servicio:

- **ej: man httpd_selinux**

Gestión de contextos

semanage - gestión de políticas
(booleanos, usuarios, login, modulo, puerto, interfaces, contexto de archivos,...)

ej:

semanage fcontext -l [-C]

-l - lista

-C - sólo reglas personalizadas

Gestión de contextos

restorecon - restaurar contexto SELinux

ej:

restorecon -Rv /var/www

-R - recursivo

-v - verbose cambios

Práctica I

Paso 1:

Crear dos archivos en /tmp

Paso 2:

Observar información de contexto

Paso 3:

Copiar uno y mover el otro a /var/www/html/

Práctica I parte 2

Paso 4:

Observar información del contexto

Paso 5:

Ver reglas definidas para la ruta destino.

Paso 6:

Restaurar la información contexto adecuada.

Práctica 2

Paso 1:

Crear un nuevo directorio, y archivo dentro.

Paso 2:

Observar información de contexto

Paso 3:

Definir un contexto para el directorio en función del uso

Práctica 2

```
semanage fcontext -a -f “” -t http_sys_content_t
‘/[nombre](/.*)?’
```

Paso 4:

Aplicar la nueva regla.

```
restorecon -RFvv /[nombre]
```

Gestión de booleanos

getsebool - recuperar el estado de un booleano

- **a - muestra todos**

semanage boolean -l

setsebool - fija el estado de un booleano

- **P - persistente**

Práctica

Permitir que el proceso httpd acceda a los directorios /home

Hacer el cambio persistente

Logging - violaciones SELinux

Rutas de log:

- /var/log/audit/audit.log
- /var/log/messages

```
sealert -l identificador
```

Práctica

Paso 1:

Crear un archivo en /root y moverlo a /var/www/html

Paso 2:

Acceder a http://localhost/[archivo]

Paso 3:

¡Investiguemos!

Mecánica de trabajo

Pasos:

- instalar
- configurar
- probar
- selinux

Casi siempre, contexto ó booleanos.

8. Hardening de servicios

8.1. SSH

SSH

- Nunca ejecutarlo en el puerto por defecto sin firewall.
  - Puerto alternativo
  - Firewall / Knockd
  - Fail2ban
- Desactivar el acceso de root mediante password.
  - Desactivarlo tambien para otros usuarios?
- Permitir el acceso solo a determinados grupos / usuarios.
- Desactivar por defecto:
  - X11 Forwarding
  - TCP Forwarding
  - GatewayPorts

SSH

- /etc/ssh/sshd_config
  - Port != 22
  - ListenAddress != 0.0.0.0
  - Protocol = 2
  - UsePrivilegeSeparation yes
  - PermitRootLogin no
  - StrictModes yes
  - PermitEmptyPasswords no
  - AllowTcpForwarding no
  - X11Forwarding no
  - GatewayPorts no
  - AllowGroups / DenyGroups
  - AllowUsers / DenyUsers

```
[...]
AllowTcpForwarding no
X11Forwarding no

AllowGroups sysadmins developers

Match Groups sysadmins
AllowTcpForwarding yes

Match Groups sysadmins developers
X11Forwarding yes
```

Apache

- Apache mod_security
  - Firewall para aplicaciones web.
    - Bloqueo de amenazas segun reglas
    - OWASP ModSecurity Core Rule Set (CRS)

9. Registros, logs.

Registro remoto

Registro remoto y alertas

- `rsyslog`
- `logcheck`

10. Suscripciones de seguridad

10.1. Listas de correo

10.2. Recursos

10.3. Recomendaciones

Listas de correo

- oss-security@lists.openwall.com
  - http://www.openwall.com/lists/oss-security/
  - The purpose of the Open Source Security (oss-security) group is to encourage public discussion of security flaws, concepts, and practices in the Open Source community. The members of this group include, but are not limited to Open Source projects, distributors, researchers, and developers.
- Listas de anuncios de seguridad de tu distro:
  - Debian: debian-security-announce@lists.debian.org
  - CentOS: centos-announce@centos.org
  - Ubuntu: ubuntu-security-announce@lists.ubuntu.com
  - Etc..

Listas de correo

- Security Bulletins de las aplicaciones en producción. Ej:
  - WikiMedia: https://lists.wikimedia.org/mailman/listinfo/mediawiki-announce
  - Drupal: https://www.drupal.org/security
  - ...
- Algunas listas a mayores:
  - bugtraq@securityfocus.com
  - fulldisclosure@seclists.org
  - ...
  - rootedcon@listas.rooted.es

Recomendaciones

- Estar pendientes de los notificaciones de seguridad del software en producción.
  - Si algún software no provee de notificaciones de seguridad siempre es útil mirar en la base de datos CVE.
    - http://www.cvedetails.com/product-list.php

- Aplicar los parches de seguridad de tu distro de forma regular.

- Desplegar sistemas de alerta.
  - logcheck
  - AIDE

11. Gestión

Bueno, ¿y ahora qué?

¿Por dónde empiezo?

Gestión de la seguridad

gestión de riesgos

identificar amenazas

calcular impacto

Roles

Analista de sistemas

Responsable de seguridad

Analista de sistemas

Identificar nuevos riesgos

Notificarlos

Ayudar para evaluar probabilidad/impacto

Proponer medidas correctivas

Responsable de seguridad

Analizar probabilidades

Informar al negocio

Establecer prioridades

Matriz de riesgos

lista de riesgos

probabilidad (1-5)

impacto (1-5)

riesgo (1-10)

medidas para mitigarlo

coste/tiempo

nuevo riesgo esperado

Ejemplos de riesgos

acceso físico a la sala de servidores

acceso remoto a los servidores web

acceso remoto a los servidores de producción

catástrofe natural

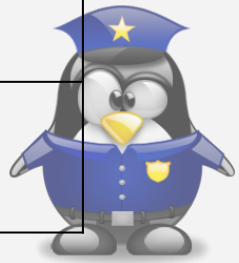
fallo eléctrico

empleado rebelde

enfermedad

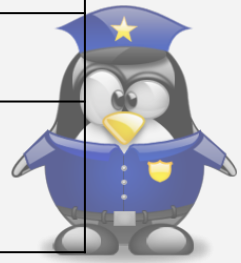
Empresa pequeña

	probabilidad	impacto	riesgo	medidas	coste	riesgo esperado
ac. físico sala servidores	4	5	9	puerta seguridad, cifrado,	5,2	2+5=7, 4+1=5
ac. remoto serv web	4	5	9	firewall, actualizaciones,...	2,1	2+5=7
ac. remoto serv produc	4	5	9	firewall, actualizaciones, dmz,	2,2,4	1+5=6, 1+5=6,
catástrofe natural	1	5	6	backup remoto	3	1+4=5



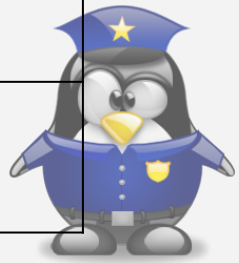
Empresa grande

	probabilidad	impacto	riesgo	medidas	coste	riesgo esperado
ac. físico sala servidores	2	5	7	seguridad 24, control biometrico,	5,3	1+5=6
ac. remoto serv web	4	5	9	firewall, actualizaciones, selinux	2,1	4+4=8
ac. remoto serv produc	3	5	9	firewall, actualizaciones, dmz,	2,2,4	1+5=6, 1+5=6,
catástrofe natural	1	5	6	centro de respaldo	5	1+1=2



Empresa productora

	probabilidad	impacto	riesgo	medidas	coste	riesgo esperado
ac. físico sala servidores	4	5	9	puerta seguridad, cifrado,	5,2	2+5=7, 4+1=5
ac. remoto serv web	4	2	6	firewall, actualizaciones,...	2,1	3+2=5
ac. remoto serv produc	4	5	9	firewall, actualizaciones, dmz,	2,2,4	1+5=6, 1+5=6,
catástrofe natural	1	5	6	backup remoto	3	1+4=5



¿Por dónde empiezo?

No depende de la tecnología.

Depende:

- situación actual
- necesidades del negocio