Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Detección de *intrusiones* en Sistemas Linux. (21390 lectures)

Per **Carlos Cortes Cortes**, ***carcoco*** (http://bulma.net/~carcoco/)

Creado el 09/12/2001 18:20 modificado el 09/12/2001 18:20

O ¿como saber si tu máquina Linux ha sido crakeada? ...

Cada día, hay más y más máquina conectadas a **Intenet**, con el consiguiente aumento en la probabilidad de sufrir un ataque por parte de **crackers**. El número de ataques reportados ha crecido espectacularmente con respecto a la cifra de años anteriores.
¿Podemos estar tranquilos y seguros detrás de nuestro potente Firewall[1]?

Linux es un sistema que potencialmente es muy **seguro**, pero necesita estar bien administrado, porque de otra forma puede ser la causa de númerosos problemas relacionados con la seguridad.

Lo peor que puede pasar es que hayan **entrado en tu máquina** y que ni te enteres, intentaré desmenuzar algunos consejos a seguir, para averiguar si nuestra máquina sigue siendo segura o no, al mismo tiempo que proporciono interesantes enlaces a documentación relacionada con este tema.

Recomendaciones de Seguridad:

1. Utilizar buenas contraseñas, para evitar los ataques a contraseña.
2. Utilizar encriptación en las comunicaciones: ssh, pgp, SSL, Ipsec ...
3. Habilitar únicamente los servicios necesarios.
4. Detectar puntos vulnerables del sistema: scanners.
5. Registro de sistema: logs.
6. Integridad del sistema: Tripwire, md5sum ...
7. Actualización del sistema debido a problemas de seguridad.
8. Utilización de Firewall.
9. Sistemas de detección de Instrusos (IDS): snort[2], LIDS[3], ...
10. Utilización de un sistema de archivos encriptado.

Aún siguiendo todas estas recomendaciones, no podremos estar seguros al 100% de que nuestro sistema no ha sido crackeado, y que cuando ejecutamos, por ejemplo, **ps -aux**, no estemos realmente ejecutando un troyano que recoge información *sensible* de nuestro sistema.
Lo que tendremos que hacer es descubrir **indicios** de la presencia de los crackers en nuestra máquina:

Verificar tamaños de archivos: **ps** suele medir 80 Kb, aunque podemos encontrar troyanos del comando **ps** de apenas 10Kb:

```
$ type ps
ps is /bin/ps
$ l /bin/ps
-r-xr-xr-x    1 root     root        83096 Dec  5 23:11 /bin/ps
```

Una precaución adicional, es tener el almacenado el resultado del comando **md5sum** de los archivos más **importantes/vulnerables**, de forma que podemos comprobar si estos han sido fraudulentamente modificados, esto se debería de hacer, después de terminar la primera instalación del sistema.

```
$ md5sum /bin/ps
3fb65605d59a7c89206926c1a600d220  /bin/ps
```

Precisamente esta es la función principal del programa **Tripwire** (http://www.tripwire.org/[4]), asegurar la integridad de los archivos, de una forma más flexible y potente que lo que tendríamos que montar a través de scripts y md5sum. Tripwire se distribuye con licencia **GPL** desde hace algun tiempo, siendo anteriormente comercial, este es el motivo por el que encontrareís varios proyectos, que se autodenominan una alternativa al **comercial** tripwire. como aide (http://www.cs.tut.fi/~rammer/aide.html[5])

Otro truco habitual de los **script kiddie** es enlazar el comando **history** de la shell, para que apunte a **/dev/null**, de forma que no registre en el historico los comandos ejecutados:

```
$ l /home/carlos/.bash_history
-rw-------    1 carlos   users        9434 Dec  8 19:44 .bash_history
```

Tendriamos que preocuparnos, si no topamos con algo como:

```
$ l /home/carlos/.bash_history
lrwxrwxrwx    1 carlos   users           9 Dec  8 19:44 .bash_history ->
/dev/null
```

Suelen usar otro tipo de **shells** mucho mas simplistas, que no registren los comandos ejecutados, de forma que se evitan el preocuparse de borrarlos. Por eso siempre es recomendable tener instalado únicamente lo que realmente se utiliza.

También suelen manipular los registros del sistema, quitando sus entradas, para que no quede constancia de sus actos en sitios/programas como **utmp**, **wtmp**, **lastlog**, **/var/log/messages** ...
Más información en Anonymizing UNIX Systems : http://www.pimmel.com/articles/anonymous-unix.html[6].
Para evitarlo:

- Mandar la parte más sensible del registro a un impresora, de forma que al cracker le seria imposible borrar estas entradas. Aunque si se da cuenta del truco, puede colapsar la impresora mandandole imprimir basura.
- Utilizar otra máquina como registro, necesitará crackear esta otro máquina para eliminar todas sus huellas.
- Mandar los logs por correo electrónico.
- Cualquier otra posibilidad que se os ocurra. Existen varios sistemas de registro de log, mucho más potentes y seguros que el syslog típico de Linux, como por ejemplo **Modular Syslog** de la empresa Core (http://sourceforge.net/projects/msyslog/[7]), con licencia BSD.

Podemos usar varios programas que detectan el borrado de estos log, algunos de los más populares son el Antizap y el Antizap2 (alguien sabe donde encontrarlos). (También podemos usar el chkrootkit, tal y como veremos más adelante)

Es interesante examinar si aparecen *misteriosamente* cuentas de usuarios desconidas en el sistema, para lo cual examinaremos el fichero **/etc/passwd**: Sólo debería aparecer el usuario **root**, con todos los privilegios.

```
$ grep :x:0: /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

Aqui tenemos la lista de todos los usuarios del sistema, si aparece alguno extraño, podemos empezar a sospechar:

```
$ cat /etc/passwd | awk -F':' '{print $1}'
```

Buscaremos en el sistema ficheros ocultos o raros, que puede ser usados para ocultar troyanos, directorios, comandos, etc...
Muchos piratas suelen crear directorios ocultos utilizando nombres como '...' (punto-punto-punto), '..' (punto-punto), '..^g' (punto-punto control+G), '\ ' (espacio en blanco), '.\ ' (punto-espacio en blanco). En algunos casos un pirata ha utilizado nombres como '.x' o '.hacker' o incluso '.mail' .

Normalmente el **script kiddie**, habrá conseguido acceso como un usuario normal y explotando algún error/bug consigue privilegios de root, los comandos/programas *setuidados* son los principales objetivos de los crackers, por lo que no es mala idea tenerlos controlados.

```
find / -perm +4000 -print
o
find / -user root -perm -4000 -print
```

Revisar la configuración de programas como **cron** y **at**, de forma que el posible pirata, no haya añadido ninguna entrada que le permita volver a entrar en el sistema posteriormente. Es interesante el averiguar realmente como ha entrado el cracker en el sistema, porque es la única forma de evitar que pueda volver a entrar de la misma forma.

Examinar el fichero /etc/inetd.conf en busca de cambios o entradas extrañas, en especial la ejecuten un shell (por ejemplo: /bin/sh o /bin/csh)

Revisar cuidadosamente ficheros relacionados con el acceso o ejecución remota de comandos, tales como, /etc/hosts.equiv, /etc/hosts.lpd y todos los .rhost del sistema.

Examinar detalladamente los ficheros de logs, analizado especialmente los logs de ftp, samba, servidor http, telnet, messages ... Nos fijaremos en entradas desde lugares extraños, verificaremos la fecha de los ficheros (es muy importante tener correctamente configurada la hora y fecha en todos los servidores).

---

chkrootkit

Los **crackers**, suelen camuflar o sustituir en **ficheros binarios** del sistema su propios ficheros troyanos, algunos de los ejemplos típicos son: login, su, telnet, netstat, ifconfig, ls, find, du, df, libc, sync, asi como los binarios listados en /etc/inetd.conf.
Nosotros tenemos que verificar que tenemos la versión original de estos ficheros y no la versión troyanizada, todo esto y mucho más lo podemos hacer usando el potente programa chkrootkit[8], en la última versión disponible detecta troyanos en todos estos **ficheros**:
aliens, asp, bindshell, lkm, rexedcs, sniffer, wted, z2, amd, basename, biff, chfn, chsh, cron, date, du, dirname, echo, egrep, env, find, fingerd, gpm, grep, hdparm, su, ifconfig, inetd, inetdconf, identd, killall, login, ls, mail, mingetty, netstat, named, passwd, pidof, pop2, pop3, ps, pstree, rpcinfo, rlogind, rshd, slogin, sendmail, sshd, syslogd, tar, tcpd, top, telnetd, timed, traceroute, write.

Siendo capaz de detectar los siguientes **RootKits**:
Solaris rootkit, FreeBSD rootkit, lrk3, lrk4, lrk5, lrk6, t0rn (and t0rn v8), some lrk variants, Ambient's Rootkit for Linux (ARK), Ramen Worm, rh[67]-shaper, RSHA, Romanian rootkit, RK17, Lion Worm, Adore Worm, LPD Worm, kenny-rk, Adore LKM, ShitC Worm, Omega Worm, Wormkit Worm, dsc-rootkit.

Una vez compilados los programas (chkwtmp, chklastlog, chkproc, chkwtmp, ifpromisc) que utiliza el chkrootkit para realizar parte de sus trabajo (chkrootkit en sí mismo es un shell script), la utilización del mismo es bastante trivial.

```
# ./chkrootkit
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not infected
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not infected
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'...

....
```

Solo puede darse un problema (de muy facil solución como vamos a ver), y que consiste en que **chkrootkit**, al ser un shell script, se basa en las siguientes herramientas:
awk, cut, echo, egrep, find, head, id, ls, netstat, ps, strings, sed, uname.

que puede contener algun troyano, la solución nos la da el propio programa, por medio de un simple parametro, le indicaremos donde debe de buscar y ejecutar esos programas que necesita:
Si los tenemos un CD-ROM:

```
./chkrootkit -p /cdrom/bin
```

En caso de tener los binarios originales de la distribución en un diskette:

```
./chkrootkit -p /floppy
```

En el caso que solo nos interese verificar algun determinado fichero lo indicaremos directamente, en este caso "ps", "login" y "sendmail".

```
# ./chkrootkit ps login sendmail
ROOTDIR is /'
Checking 's'... not infected
Checking 'login'... not infected
Checking 'sendmail'... not infected
```

Aqui tenemos todas las opciones disponibles en este interesante programa:

```
# ./chkrootkit --help
Usage: ./chkrootkit [options] [test ...]
Options:
        -h               show this help and exit
        -V               show version information and exit
        -l               show available tests and exit
        -d               debug
        -q               quiet mode
        -x               expert mode
        -r dir           use dir as the root directory
        -p dir1:dir2:dirN path for the external commands used by chkrootkit
```

Para verificar la alteración de los logs, usaremos los programas chkwtmp y chklastlog:

```
# ./chkwtmp

# ./chklastlog
```

Pudiendo testear incluso si la interfaz esta en modo promiscuo, lo que indicaria que tenemos algun sniffer funcionando:

```
./ifpromisc
ppp0 is not promisc
```

Referencias adicionales:

- How to tell if your Linux box has been cracked: http://www.linuxworld.com/site-stories/2001/1012.cracked.html[9]
- Unix Incident Guide: How to Detect an Intrusion. http://www.ciac.org/ciac/documents/CIAC-2305_UNIX_Incident_Guide_How_to_Detect_an_Intrusion.pdf[10]
- Detección de Intrusos 1.0 (por J.J.F Team): http://www.aebius.com/b_datos_doc/archivos/deteccion_intrusos.htm[11]
- CERT's Detecting Signs of Intrusion: http://www.cert.org/security-improvement/modules/m09.html[12]
- CERT's http://www.cert.org/tech_tips/unix_configuration_guidelines.html[13]
- CERT's http://www.cert.org/tech_tips/intruder_detection_checklist.html[14]
- CERT's http://www.cert.org/tech_tips/root_compromise.html[15]
- FAQ: Network Intrusion Detection System. http://www.robertgraham.com/pubs/network-intrusion-detection.html
- Help when broken into: http://www.fish.com/tct/help-when-broken-into[16]
- Computer Forensics Analysis Class Handouts: http://www.fish.com/forensics/class.html[17]
- Intrusion Detection FAQ: http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm[18]
- chkrootkit (locally checks for signs of a rootkit): http://www.chkrootkit.org/[19]

- RID (remote intrusion detector): http://www.theorygroup.com/Software/RID/[20]
- Know Your Enemy: A Forensic Analysis. http://project.honeynet.org/papers/forensics/[21]
- The DoS Project's "trinoo" distributed denial of service attack tool: http://staff.washington.edu/dittrich/misc/trinoo.analysis[22]
- Know Your Enemy: Passive Fingerprinting. http://project.honeynet.org/papers/finger/[23]
- "Root Kits" and hiding files/directories/processes after a break it: http://staff.washington.edu/dittrich/misc/faqs/rootkits.faq[24]
- Charla "Network Intrusion Detection Systems. Snort based NIDS": http://umeet.uninet.edu/conferencias/05-12-2000/0512.html[25]
- Preparing to Detect Signs of Intrusion: http://www.arcert.gov.ar/webs/textos/intrusion.pdf[26]
- Intrusion Detection Methodologies: http://www.arcert.gov.ar/webs/textos/idmethods.pdf[27]

No estaría mal que alguien nos contara sus experiencias personales tanto positivas como negativas en este *interesante* campo. (Ricardo, Ricardo estas ahi?)

--
Carlos Cortes(aka carcoco)
http://bulma.net/todos.phtml?id_autor=132 [28]

Lista de enlaces de este artículo:

1. http://bulma.net/ body.phtml?nIdNoticia=861
2. http://www.snort.org
3. http://www.lids.org
4. http://www.tripwire.org/
5. http://www.cs.tut.fi/~rammer/aide.html
6. http://www.pimmel.com/articles/anonymous-unix.html
7. http://sourceforge.net/projects/msyslog/
8. http://www.chkrootkit.org/
9. http://www.linuxworld.com/site-stories/2001/1012.cracked.html
10. http://www.ciac.org/ciac/documents/CIAC-2305_UNIX_Incident_Guide_How_to_Detect_a
11. http://www.aebius.com/b_datos_doc/archivos/deteccion_intrusos.htm
12. http://www.cert.org/security-improvement/modules/m09.html
13. http://www.cert.org/tech_tips/unix_configuration_guidelines.html
14. http://www.cert.org/tech_tips/intruder_detection_checklist.html
15. http://www.cert.org/tech_tips/root_compromise.html
16. http://www.fish.com/tct/help-when-broken-into
17. http://www.fish.com/forensics/class.html
18. http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm
19. http://www.chkrootkit.org/
20. http://www.theorygroup.com/Software/RID/
21. http://project.honeynet.org/papers/forensics/
22. http://staff.washington.edu/dittrich/misc/trinoo.analysis
23. http://project.honeynet.org/papers/finger/
24. http://staff.washington.edu/dittrich/misc/faqs/rootkits.faq
25. http://umeet.uninet.edu/conferencias/05-12-2000/0512.html
26. http://www.arcert.gov.ar/webs/textos/intrusion.pdf
27. http://www.arcert.gov.ar/webs/textos/idmethods.pdf
28. http://bulma.net/todos.phtml?id_autor=132

E-mail del autor: carcoco _ARROBA_ gmail.com