

# Bases de seguridad

## 1. Seguridad informática

Los principales objetivos de la seguridad informática conciernen a:

- ✓ **La seguridad de la conexión:** se trata de controlar que los usuarios que se conectan dispongan efectivamente de la autorización para ello y de prohibirles el acceso al sistema en caso contrario.
- ✓ **La integridad de los datos:** se trata de conseguir que los archivos y las bases de datos no estén corruptas y de mantener la coherencia entre los datos.
- ✓ **La confidencialidad de los datos:** el acceso a los datos para consulta y modificación se debe limitar únicamente a usuarios autorizados.

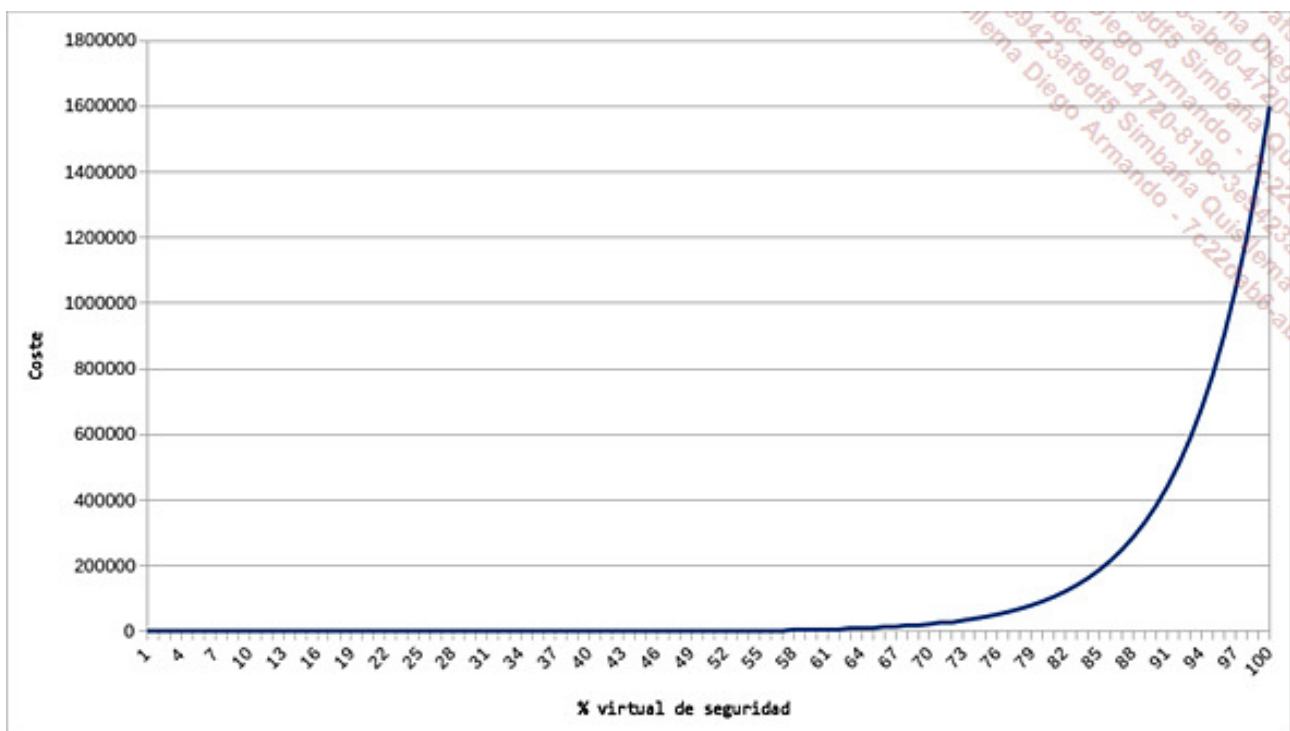
Dispone de varios medios:

- ✓ La autenticación de los usuarios mediante contraseña.
- ✓ El cifrado de los datos.
- ✓ La seguridad física controlando el acceso de las personas a las salas informáticas, mediante circuitos físicos inviolables.
- ✓ La información relativa a los riesgos penales en los que se incurre en caso de infracción. Un "atracó" informático es un delito, no un juego.
- ✓ El control frecuente de los permisos de acceso a los archivos y bases de datos.
- ✓ El control de acceso a los servidores y al software.
- ✓ El control de los «checksum» de los archivos para asegurar su integridad.
- ✓ La copia de seguridad regular de los datos.
- ✓ El control de los principales eventos del sistema.
- ✓ La instalación de cortafuegos (*firewall*) que controlan los accesos al sistema informático desde el exterior y evitan que los usuarios accedan a servicios externos sin querer o sin necesitarlo y, de esa manera, limitar el riesgo de propagación de virus.
- ✓ El uso de firewall de aplicaciones para analizar el tráfico de datos, por ejemplo

para detectar los ataques a los servidores web.

- ✓ El uso de herramientas de detección y prevención de intrusiones (por correlación de trazas) y el filtrado de direcciones o personas pertinentes.
- ✓ El uso de puntos de control para un mejor filtrado de los accesos.
- ✓ La instalación de un antivirus, incluso en Linux, si el servidor trata datos desde sistemas operativos susceptibles de tener virus y hacia ellos.
- ✓ La instalación de herramientas antispams y antispyswares, según el mismo principio, con el fin de evitar una intrusión y la saturación de los servidores de correo electrónico.
- ✓ Iniciar únicamente los servicios realmente útiles en el servidor y el cliente.
- ✓ Y muchos otros...

La seguridad es también una cuestión de la relación coste/riesgo. Los costes no son proporcionales al nivel deseado, son más bien exponenciales. El gráfico siguiente expone de manera simplificada el problema:



*Informe coste/eficacia de la seguridad*

La seguridad perfecta no existe. Cuanto mayor sea el nivel de seguridad que se deba

obtener, mayor será su coste. Esto implica el tiempo para configurarlo, los productos hay que comprar, las personas hay que emplear y formar.... Debemos encontrar un punto intermedio, usando el sentido común, y por último ser un poco peor que el pirata o el hacker básico. ¿Qué quiere proteger? ¿Opera con datos personales, sensibles, bancarios, de bolsa? ¿O es un simple sitio internet público sin datos sensibles ni acceso a su sistema de información? ¿Debemos caer en la paranoia?

Aquí nos limitaremos a la seguridad que los componentes del sistema nos pueden ofrecer.

Algunos métodos sencillos permiten limitar los riesgos de acceso al sistema, sin un coste significativo:

- ˘ Puede definir un valor de umask restrictivo (p. ej.: 077) para extender a continuación los permisos de acceso de algunos archivos.
- ˘ No debe apartarse del terminal sin desconectarse o bloquearlo (una buena broma en caso contrario consiste en utilizar el cliente de correo de la persona en cuestión para que le traigan una pizza; después de algún tiempo, el resultado es radical...).
- ˘ Hay que prestar atención a las fechas de los últimos inicios de sesión logrados e infructuosos que aparecen en cada conexión.
- ˘ No permitir nunca el acceso, incluso en modo de sólo lectura, a los archivos de sesión como .profile.
- ˘ Nunca poner el "." en primera posición del PATH, y controlar sus rutas.
- ˘ Si los servicios lo autorizan, emplee un entorno de tipo chroot.
- ˘ Si es posible, piense en compartimentar (containers docker).
- ˘ Verifique las fuentes del software y las firmas de los paquetes.
- ˘ Evite fuentes de instalación dudosas.
- ˘ Evite las herramientas de crack y hack (generadores de números de serie...) vectores de virus, spyware y otros.
- ˘ Utilice un antivirus, de igual forma en Linux.
- ˘ Verifique con regularidad los logs de acceso a los equipos.
- ˘ Actualice su distribución regularmente.
- ˘ Active el firewall por defecto.

- ✓ Utilice una contraseña fuerte.
- ✓ Favorezca los protocolos seguros (SSH, HTTPS...).
- ✓ Utilice claves complejas.

## 2. Controlar los privilegios especiales

Los privilegios especiales de ejecución (bits SUID y SGID) suelen ser causa de inseguridad en el sistema. En efecto, un usuario malintencionado, aprovechando la falta de atención o la ausencia de un compañero o un administrador que no está desconectado de su consola, puede modificar los permisos de ciertos comandos a su favor. El ejemplo más habitual es el de reescribir un shell como un programa poco usado (por ejemplo `sx`) y darle los privilegios SUID. Al iniciar este comando, se puede convertir en root.

Obtener el permiso de listar todos los archivos:

```
# chmod u+s cat
```

Obtener un shell root:

```
# cp /bin/sh /bin/sx
# chmod u+s /bin/sx
...
$ sx
# ...
```

El comando siguiente permite buscar todos los archivos que disponen de los bits SUID o SGID:

```
# find / -type f \( -perm -4000 -o -perm -2000 \)
# find / -type f \( -perm -4000 -o -perm -2000 \)
/bin/su
/bin/umount
/bin/eject
```

```

/bin/mount
/bin/ping
/bin/ping6
/sbin/unix_chkpwd
/sbin/unix_chkpwd
/usr/bin/expiry
/usr/bin/write
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/gnatski
/usr/bin/mahjongg
/usr/bin/chfn
/usr/bin/yset
/usr/bin/wall
/usr/bin/crontab
/usr/bin/v4l-conf
/usr/bin/gnome
/usr/bin/same-gnome
/usr/bin/gnotravex
/usr/bin/gnobot2
...

```

En la lista anterior, hay un intruso: **/usr/bin/yset** , que permite modificar las configuraciones de un servidor de sonido y que no necesita para nada disponer del privilegio SUID. Hay un problema.

```

# ls -l /usr/bin/yset
-rwsr-sr-x 1 root root 604040 may 19 21:28 /usr/bin/yset
# md5sum /usr/bin/yset
04ff72010ff1cf1c14d7706159cdf8bf /usr/bin/yset
# ls -l /bin/bash
-rwxr-xr-x 1 root root 604040 jan 22 2016 /bin/bash
# md5sum /bin/bash
04ff72010ff1cf1c14d7706159cdf8bf /bin/bash

```

Alguien ha vuelto a escribir un shell con otro nombre. Éste es un caso experimentado por el autor.

### 3. Comprobar los paquetes

El capítulo Instalación de Linux y de los paquetes de software ha tratado sobre toda la gestión de los paquetes de software. Entre las diversas opciones, algunas permiten controlar la autenticidad de un paquete. El sistema de paquetería RPM contiene, además del nombre del archivo, su tipo (configuración, binario, etc.) y en algunos casos (binario) la suma de control (checksum) MD5 del archivo. No existe un comportamiento equivalente para dpkg.

Según el ejemplo anterior, ¿cómo restaurar el archivo yset? En tres etapas:

→ Encontrar el paquete de origen:

```
# rpm -qf /usr/bin/yset
yiff-2.14.5-0.pm.1
```

→ Controlar el estado del paquete instalado:

```
# rpm -V yiff
SM5....T /usr/bin/yset
```

- ˘ S: el tamaño no es el correcto
- ˘ M: se han modificado los permisos
- ˘ 5: la suma de control MD5 es diferente
- ˘ T: la fecha de modificación no es la correcta.

→ Vuelva a instalar el paquete de origen según las modalidades propias de su distribución.

### 4. Política de la contraseñas

Las contraseñas son la base de la autenticación de un usuario. Deben ser seguras. Sin

embargo, suele ser la asignatura pendiente, tanto en el trabajo como en casa, e incluso en Internet:

- ✓ contraseña escrita en un post-it;
- ✓ uso de un gestor de contraseña automático, a su vez sin contraseña;
- ✓ misma contraseña para todos los sitios web y software;
- ✓ contraseña nunca cambiada;
- ✓ misma contraseña o cuenta para toda la familia/servicio;
- ✓ contraseña demasiado sencilla;
- ✓ etc.

No sirve de nada caer en la paranoia. Tiene que encontrar un término medio. Si pide a los usuarios que modifiquen su contraseña demasiado a menudo, o si es demasiado difícil, tienden a apuntarla. Si es demasiado fácil y deja pasar demasiado tiempo, ya no es seguro.

Los usuarios deben elegir una buena contraseña, evitando la sencillez o más bien lo evidente: nombres de los hijos, de la esposa, de lugares, fecha de nacimiento y, en general, todo lo que importa y que es conocido del entorno profesional o personal.

Un término medio puede ser modificar las reglas de cambio de contraseña con `chage` (o `passwd`), de forma que se establezca una duración para la validez de la contraseña de 40 días. Se han presentado los comandos de modificación de la política de gestión de las contraseñas en el capítulo Las tareas administrativas - Administración de los usuarios.

```
root@ubuntu:~# chage -l jolivares
Último cambio de contraseña          : ago 04, 2017
La contraseña caduca                  : mar 18, 2018
Contraseña inactiva                   : mar 18, 2018
La cuenta caduca                      : ene 01, 2019
Número de días mínimo entre cambio de contraseña : 7
Número de días máximo entre cambio de contraseña : 4
Número de días de aviso antes de que caduque la contraseña : 10
```

Los módulos PAM influyen en la política de gestión de las contraseñas, obligando en algunos casos a elegir uno más o menos complejo. Aunque parezca una paradoja, una

contraseña debe ser fácil de recordar por un usuario, lo que no implica forzosamente que sea fácil de piratear (por John the Ripper, por ejemplo). Existen contraseñas que se pueden recordar por medios mnemotécnicos. También puede generar contraseñas de manera automática con la herramienta **pwgen**.

```
$ pwgen
```

```
uash6She lohJo7ae Ohphab3i ouRik9ie uM4va3im Neer7Eit eib3Hauy xo9luy5p
ahSiW0uf AhG6wail Yai6neeh phae4ioV deeL3aip Uz5ahzaa aiV5phee Aegaity7x
ioPh1ahn Ong6Baib Eish4rip eik9Gie1 ien3Iepe xohduj7U aiP2keov So5ovaht
Voh9oxoe ahs2Meeg Ooch5xix Phe3yiuZ eeCa5ohv aig9Ai3o Go4Ateeh Hee6thei
Rai6Daeh aid8ieNg Thah6ien daphaiG0 Iefai5oh Pheife6i Poora8ah Coh5Aida
ViC7ieth hohG5sei Aa9Jeilu eopoX8Si jooh3Eif dooPhai1 chohqu1G ieNgae3o
wiCeisi3 aej6PieV eoTha1Fu ieR2yeeb Eireili6 saiGhie2 XohRoo1a cahb2Yah
Guungah0 ube3vo0D oshol3Op Pui6agh5 Ao7baeN1 foTek9Ei aeM3lala Ene2baol
geloV9ai Weeyu2ie Uvae2Vie dei0euL7 Xee9uaza ed8Eeghu eebiu2Ka zey0Lih
be6Ailoi eiph8Ohb Yahpahr4 aij4dahG oQu2chae Fe5eeg9c Hoosh6oh lip8eiwe
AuPie0um Ahxai9eo Dae5oquu le7Viek8 pa2aew8B fohham7A fah1Oogi ieH9vee8
```

Estas contraseñas son pseudoaleatorias. Si habla inglés (y geek/leet), estas contraseñas representan una pronunciación. Por ejemplo:

```
dooPhai1: Do you fail?
```

Puede pedir que se generen contraseñas totalmente aleatorias con una longitud dada, en este caso de 16 caracteres con al menos un carácter especial:

```
$ pwgen -sy -l 16
"%Eie*0s3KKUa_@T
```

## 5. Almacenar las contraseñas

¡Ni se le ocurra ponerlas bajo el teclado ni en un post-it pegado a la pantalla! Usted no pone las llaves de su casa debajo de la alfombra o en la maceta que hay junto a la puerta, ¿no es verdad?



Se vuelve complicado recordar todas las contraseñas tanto de los servidores como de los servicios en línea (foros, sitios de comercio electrónico, redes sociales...). Existen herramientas para ayudarle. Los administradores de contraseñas, llamados Password Managers, llaveros, portallaves, monederos, pueden almacenar de forma segura todas sus contraseñas en un solo lugar, a través de una clave, una contraseña maestra o un archivo codificado.

Por lo general, podemos considerar los llaveros proporcionados por los sistemas operativos como relativamente seguros, por supuesto con la condición de que su formato sea codificado, que proporcione una contraseña y que su sesión sea protegida de la misma forma. KDE o Gnome proporcionan estos llaveros.

En entornos empresariales, contar con un puesto de trabajo en Linux es todavía poco frecuente. Se trata por lo general de puestos Windows o Mac OS X. En ese caso, las herramientas de tipo keepass y sus derivados (**keepassx** por ejemplo) son una buena alternativa ya que están disponibles para todos los sistemas operativos.

## 6. Prohibir las conexiones

### a. **/bin/false**

Algunas cuentas no deben ser interactivas: se deben prohibir las conexiones desde una consola. Se pueden asignar estas cuentas a una aplicación, a un servicio, a una conexión FTP, etc., pero se debería rechazar la conexión: ¡no shell!

En la lista de los shells autorizados, uno llama la atención:

```
$ cat /etc/shells
/bin/ash
/bin/bash
/bin/bash1
/bin/csh
/bin/false
/bin/ksh
/bin/sh
/bin/tcsh
```

```

/bin/true
/bin/zsh
/usr/bin/csh
/usr/bin/ksh
/usr/bin/passwd
/usr/bin/bash
/usr/bin/tcsh
/usr/bin/zsh

```

El shell `/bin/false` no es realmente un shell. Prohíbe las conexiones interactivas. Ya hemos encontrado este comando **false** en el capítulo El shell y los comandos GNU: devuelve siempre falso. En cuanto **login** intenta ejecutar el shell de conexión, se rechaza al usuario.

El comando `/sbin/nologin` hace casi la misma cosa, mostrando el contenido del archivo `/etc/nologin.txt` si existe.

```

# egrep "(false|nologin)$" /etc/passwd
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
...

```

## b. `/etc/nologin`

Antes de pasar por un pseudoshell de conexión, los módulos PAM permiten aplicar numerosas limitaciones. Entre ellas, el módulo **pam\_nologin**, que le permite prohibir la conexión de los usuarios excepto root. Este módulo monitoriza si un usuario intenta conectarse al archivo `/etc/nologin`. Es útil para tareas de mantenimiento, ya que sólo root se puede conectar.

```
# grep nologin /etc/pam.d/*
# grep nologin /etc/pam.d/*
/etc/pam.d/gdm-autologin:auth      requisite    pam_nologin.so
/etc/pam.d/gdm-fingerprint:auth   requisite    pam_nologin.so
/etc/pam.d/gdm-launch-environment:auth requisite    pam_nologin.so
/etc/pam.d/gdm-password:auth      requisite    pam_nologin.so
/etc/pam.d/login:auth             requisite    pam_nologin.so
/etc/pam.d/ppp:auth               required     pam_nologin.so
/etc/pam.d/sshd:account            required     pam_nologin.so
```

Piense también que puede prohibir el acceso de una cuenta dada vía el módulo PAM **pam\_listfile**.

### c. `/etc/securetty`

En el mismo estilo, el archivo `/etc/securetty` contiene la lista de los terminales considerados como seguros. Para el servicio dado, se prohibirá la conexión si el terminal de la persona que intenta conectarse no está incluido. En el ejemplo siguiente, se autorizan los logins (mediante el comando `login`) únicamente desde los seudoterminales locales, es decir, sólo desde las consolas directamente accesibles en el ordenador mediante las combinaciones `[Alt][F1]` a `[Alt][F7]`.

```
# grep securetty /etc/pam.d/*
/etc/pam.d/login:auth [success=ok new_authtok_reqd=ok ignore=ignore
user_unknow=bad default=die] pam_securetty.so

# cat /etc/securetty | wc -l
412
# grep tty /etc/securetty
tty1
tty2
tty3
tty4
tty5
...
```



¡No confunda securetty (Secure tty) con security!

## 7. Probar las contraseñas

Las herramientas crack y John the ripper intentan descifrar sus contraseñas, tanto desde un diccionario como por fuerza bruta (unas tras otras). En el peor de los casos, las encuentran en unos segundos; en el mejor, en varios días, incluso semanas. Si esto es así, se puede considerar que la contraseña es buena.

Probablemente deberá recompilar John the ripper desde las fuentes. He aquí como hacerlo en Fedora 31 o CentOS 8, como root, para la versión 1.9.0:

```
# yum -y install wget gpgme
# yum -y group install "Development Tools"
# cd
# wget https://www.openwall.com/john/k/john-1.9.0.tar.z
# wget https://www.openwall.com/john/k/john-1.9.0.tar.gz.sign
# wget https://www.openwall.com/signatures/openwall-offline-signatures.asc
# gpg --import openwall-offline-signatures.asc
# gpg --verify john-1.9.0.tar.gz.sign
# tar xvfzf john-1.9.0.tar.gz
# cd john-1.9.0/src
# make clean linux-x86-64
# cd ../run/
# ./john --test
Benchmarking: descript, traditional crypt(3) [DES 128/128 SSE2]... DONE
Many salts: 5064K c/s real, 5167K c/s virtual
Only one salt: 4855K c/s real, 4904K c/s virtual
...

# wget -O - https://mirrors.kernel.org/openwall/wordlists/all.gz |
gunzip -c > openwall.dico
```

John the ripper es muy fácil de usar empleando el comando **john**. Por supuesto, no se puede probar de forma correcta sin ser root. Para probar la integridad de su archivo,

/etc/shadow :

```
# ./john /etc/shadow
Loaded 2 password hashes with 2 different salts (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:07 60% 1/3 0g/s 394.3p/s 394.3c/s 394.3C/s R9999902..root9999988
0g 0:00:00:08 65% 1/3 0g/s 395.5p/s 395.5c/s 395.5C/s R9999986..99999N
```

En el modo por defecto, john:

- ˘ intenta una detección simple vía combinaciones corrientes relacionadas con la cuenta;
- ˘ pasa al modo diccionario con aplicación de reglas;
- ˘ luego, intenta una búsqueda incremental.

¡Una búsqueda puede tardar de unos segundos a varias semanas!

Para probar un solo usuario:

```
# john -user:seb /etc/shadow
```

Para probar los usuarios con un diccionario:

```
# ./john -users:seb -wordlist: ./openwall.dico /etc/shadow
```

Lo mismo pero probando varias reglas por palabra (inversión, mayúsculas, minúsculas, etc.):

```
# ./john -users:seb -wordlist: ./openwall.dico -rules /etc/shadow
```

Para retomar una búsqueda interrumpida ([Ctrl] C):

```
# john -restore
```

John ubica sus resultados en el directorio ~/.john. Generalmente sólo root debe utilizar esta herramienta:

```
# ls -l .john
total 4
-rw----- 1 root root 70 may 23 21:58 john.pot
-rw----- 1 root root 124 may 23 21:54 john.rec
```

El archivo `john.pot` contiene los resultados encontrados por John. El archivo no está vacío y esto indica que hay un problema: john ha encontrado una contraseña. El archivo `john.rec` contiene el estado actual de la búsqueda por si se produce una interrupción y es utilizado por el programa para reanudar la búsqueda.

En una máquina basada en Intel Core 2 duo de 64 bits a 3,4 GHz, se craqueó la contraseña de Seb (el autor) en 4 minutos y 22 segundos. Estaba (de manera voluntaria esta vez) basada en una palabra del diccionario.

Si pulsa una tecla durante la búsqueda, John muestra su estado.

Imaginemos una cuenta Enrique cuyo propietario tiene como contraseña la misma palabra que su nombre de cuenta, a saber "Enrique":

```
# john -users:enrique /etc/shadow
# ./john -users:enrique /etc/shadow
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
enrique      (enrique)
1g 0:00:00:00 100% 1/3 2.222g/s 213.3p/s 213.3c/s 213.3C/s Enrique..h999998
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Eso debería hacer reflexionar a los que piensan que nadie vendrá a molestarle y debería animarlos a modificar su contraseña con reglas precisas (mayúsculas, minúsculas, cifras,

etc.).

## 8. Buscar rootkits

### a. Fundamentos del rootkit

Una vez que un pirata informático haya conseguido, mediante un fallo o una contraseña demasiado sencilla, penetrar en su máquina, seguramente tratará de abrirse una puerta de entrada mayor, o más bien una puerta trasera, un **backdoor**, para así volver a utilizar su máquina con propósitos dudosos, o llevarse o almacenar datos (de esta manera, algunos PC se han encontrado con miles de archivos de audio o de video, o han sido utilizados como plataforma de ataque (PC zombi) por personas poco escrupulosas).

Lo ideal para el pirata es poder acaparar los derechos de root. Es su objetivo: ser el dueño total de su máquina. Para ello, no siempre le hace falta «teclear» directamente en esta cuenta. En Linux, es habitual (e incluso aconsejable) conectarse como simple usuario y luego cambiar a root mientras se efectúan las operaciones necesarias. Ahora bien, para pasar a root, puede utilizar **su** o **sudo**.

Si, mediante el servicio FTP mal configurado o mediante ssh (y luego scp), el pirata intenta conectarse a una cuenta cuya contraseña es obvia (como en el ejemplo de la cuenta enrique), y consigue desplegar en el sistema un script llamado «su» y modifica el PATH por defecto para poner en él la ruta al script, entonces, es pan comido.

```
$ pwd
/home/seb
$ cat su
#!/bin/bash
echo -e "Contraseña :\c"
read -s password
echo "$@ $password" > /tmp/fic
echo
echo "su: Contraseña incorrecta."
/bin/su $@
$ chmod +x su
$ export PATH=$HOME:$PATH
```

```
$ su - root
Contraseña: ==> FALSO SU
su: Contraseña incorrecta. ==> FALSO SU
Contraseña: ==> VERDADERO SU
```

Sólo falta mostrar el contenido del archivo para obtener la contraseña:

```
$ cat /tmp/arc
- root azerty
```

El método, simplista pero eficaz, no es imparable ni mucho menos: su sólo pide la contraseña una vez, y a no ser que se sea muy despistado, el engaño salta a la vista. Al hecho de ubicar scripts, modificar el entorno, sustituir un archivo por otro de modo que se obtenga un acceso privilegiado a una máquina, se le denomina instalar un **rootkit**. Una vez en su sitio, este último garantiza, mientras no se detecte, un acceso root a la máquina.

## b. Chkrootkit y rkhunter

La herramienta **chkrootkit** es una herramienta sencilla que permite buscar la presencia de los rootkits más conocidos y habituales. Sólo es eficaz si se actualiza y se ejecuta regularmente; no sustituye a los controles ya nombrados anteriormente.

```
# chkrootkit
ROOTDIR is '/'
Checking 'amd'... not found
Checking 'basename'... not infected
Checking 'biff'... not found
Checking 'chfn'... not infected
Checking 'chsh'... not infected
Checking 'cron'... not infected
Checking 'crontab'... not infected
...
Searching for sniffer's logs, it may take a while... nothing found
Searching for HiDrootkit's default dir... nothing found
Searching for t0rn's default files and dirs... nothing found
Searching for t0rn's v8 defaults... nothing found
```



```

Searching for Lion Worm default files and dirs... nothing found
Searching for RSHA's default files and dir... nothing found
Searching for RH-Sharpe's default files... nothing found
Searching for Ambient's rootkit (ark) default files and dirs... nothing found
Searching for suspicious files and dirs, it may take a while...
...
Searching for LPD Worm files and dirs... nothing found
Searching for Ramen Worm files and dirs... nothing found
Searching for Maniac files and dirs... nothing found
Searching for RK17 files and dirs... nothing found
Searching for Ducoci rootkit... nothing found
Searching for Adore Worm... nothing found
Searching for ShitC Worm... nothing found
Searching for Omega Worm... nothing found
Searching for Sadmin/IIS Worm... nothing found
...

```

**Rkhunter** es todavía más potente. Es capaz de actualizarse solo, en particular para recuperar archivos de firmas, como lo haría un antivirus.

```

# rkhunter --update
[ Rootkit Hunter version 1.4.2 ]

Checking rkhunter data files...
Checking file mirrors.dat           [ Updated ]
Checking file programs_bad.dat      [ Updated ]
Checking file backdoorports.dat     [ No update ]
Checking file suspscan.dat          [ Updated ]
Checking file i18n/cn               [ No update ]
Checking file i18n/de               [ Updated ]
Checking file i18n/en               [ No update ]
Checking file i18n/tr               [ Updated ]
Checking file i18n/tr.utf8          [ Updated ]
Checking file i18n/zh               [ Updated ]
Checking file i18n/zh.utf8          [ Updated ]

```

La lista de pruebas que rkhunter puede efectuar puede consultarse empleando **--list**. Ejecute una verificación completa con **-checkall**. La lista de pruebas efectuadas es muy extensa para ser copiada aquí, pero un resumen se muestra al final:

```
System checks summary
=====
File properties checks...
  Required commands check failed
  Files checked: 135
  Suspect files: 4
Rootkit checks...
  Rootkits checked : 503
  Possible rootkits: 0
Applications checks...
  All checks skipped
The system checks took: 3 minutes and 58 seconds
All results have been written to the log file: /var/log/rkhunter/rkhunter.log
One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter/rkhunter.log)
```

El programa cree haber encontrado archivos sospechosos; ahora debemos estudiar el archivo de log.

```
[22:10:19] Warning: Checking for prerequisites          [ Warning ]
[22:10:19] Warning: WARNING! It is the users responsibility to ensure
that when the '--propupd' option
[22:10:25] /usr/bin/egrep                                [ Warning ]
[22:10:25] Warning: The command '/usr/bin/egrep' has been replaced by a
script: /usr/bin/egrep: a /usr/bin/sh script, ASCII text executable
[22:10:25] /usr/bin/fgrep                                [ Warning ]
[22:10:25] Warning: The command '/usr/bin/fgrep' has been replaced by a
script: /usr/bin/fgrep: a /usr/bin/sh script, ASCII text executable
[22:10:32] /usr/libexec/nm-ifdown                         [ Warning ]
[22:10:32] Warning: The command '/usr/libexec/nm-ifdown' has been replaced by a
script: /usr/libexec/nm-ifdown: /usr/bin/sh script, ASCII text executable
[22:10:32] /usr/libexec/nm-ifup                          [ Warning ]
[22:10:32] Warning: The command '/usr/libexec/nm-ifup' has been replaced by a
```

```
script: /usr/libexec/nm-ifup: a /usr/bin/sh script, ASCII text executable
[22:14:07] Checking for passwd file changes          [ Warning ]
[22:14:07] Warning: Unable to check for passwd file differences: no copy
of the passwd file exists.
[22:14:07] Checking for group file changes           [ Warning ]
[22:14:07] Warning: Unable to check for group file differences: no copy of the
group file exists.
```

Una opción importante es **--propupd**. Esta se utiliza en la postinstalación del sistema. Creará un archivo **rkhunter.dat** que servirá de base de comparación para las siguientes pruebas.

## 9. Los virus

Ya han aparecido los primeros virus en Unix. Aunque se suele dotar de mayor seguridad al sistema y aunque los virus en las plataformas Unix y Linux (incluyendo Mac OS X) son casi inexistentes, en el caso más o menos probable de que un virus penetre y comprometa la seguridad de su máquina, las de otros o la de toda la red, tiene la responsabilidad de erradicarlo.

Si su máquina es un servidor, en particular de correo electrónico o de archivos en una red que contiene máquinas con Windows muy expuestas, no debe servir de vector indirecto de propagación. Tiene que eliminar la amenaza.

Existe varios antivirus en Linux, algunos comerciales (gratuitos o no), otros libres. El antivirus Clam (ClamAV) es libre y gratuito. También es uno de los que ofrece mejor rendimiento. Está disponible en la dirección <http://www.clamav.net/>. Se actualiza cada día.

El comando **freshclam** permite actualizar las bases de firmas, ubicadas en **/var/lib/clamav** :

```
# pwd
/var/lib/clamav
# freshclam
```

```

ClamAV update process started at Feb 11 22:23:06 2020
WARNING: Your ClamAV installation is OUTDATED!
WARNING: Local version: 0.101.5 Recommended version: 0.102.2
DON'T PANIC! Read https://www.clamav.net/documents/upgrading-clamav
Downloading main-59.cdiff [100%]
main.cld updated (version: 59, sigs: 4564902, f-level: 60, builder: sigmgr)
Downloading daily-25643.cdiff [100%]
Downloadingdaily-25644.cdiff [100%]
...
daily.cld updated (version: 25720, sigs: 2181998, f-level: 63, builder: raynman)
bytecode.cvd is up to date (version: 331, sigs: 94, f-level: 63, builder: anvilleg)
Database updated (6746994 signatures) from database.clamav.net (IP: 2606:4700::6810:db54)
# II
total 476564
-rw-r--r--. 1 clamupdate clamupdate 296388 19 sep 18:12 bytecode.cvd
-rw-r--r--. 1 clamupdate clamupdate 180292608 11 feb 22:23 daily.cld
-rw-r--r--. 1 clamupdate clamupdate 307403264 11 feb 22:23 main.cld
-rw-----. 1 clamupdate clamupdate 64 11 feb 22:23 mirrors.dat

```

**clamscan** permite buscar los posibles virus:

```

# clamscan -v /usr/bin
...
----- SCAN SUMMARY -----
Known viruses: 6736784
Engine version: 0.101.5
Scanned directories: 1
Scanned files: 1759
Infected files: 0
Data scanned: 256.00 MB
Data read: 87.50 MB (ratio 0.85:1)
Time: 13.742 sec (0 m 26 s)

```

A continuación, la misma prueba con un virus falso bajo tres formas: binario, comprimido gzip y comprimido bzip2. Si se detecta un virus, se mueve el archivo correspondiente a **/home/seb/VIRUS** :

```

$ clamscan -v -r --move=/home/seb/VIRUS /home/seb/bin
Scanning /home/seb/bin/eicarcom2.zip
/home/seb/bin/eicarcom2.zip: Eicar-Test-Signature FOUND
/home/seb/bin/eicarcom2.zip: moved to '/home/seb/VIRUS//eicarcom2.zip'
Scanning /home/seb/bin/eicar_com.zip
/home/seb/bin/eicar_com.zip: Eicar-Test-Signature FOUND
/home/seb/bin/eicar_com.zip: moved to '/home/seb/VIRUS//eicar_com.zip'
Scanning /home/seb/bin/eicar.com
/home/seb/bin/eicar.com: Eicar-Test-Signature FOUND
/home/seb/bin/eicar.com: moved to '/home/seb/VIRUS//eicar.com'

----- SCAN SUMMARY -----
Known viruses: 1026485
Engine version: 0.97
Scanned directories: 1
Scanned files: 3
Infected files: 3
Data scanned: 0.00 MB
Time: 1.216 sec (0 m 1 s)

```

Se puede iniciar Clamav como servicio. En este caso, se puede configurar la búsqueda de virus en estructuras concretas, lo que permite la existencia de herramientas como clamtk o la integración de funciones en gestores de archivos (como Nautilus en Gnome), o si existe un cliente de mail como Thunderbird.

## 10. Los límites del usuario

El campo de acción de los PAM es más amplio que la simple conexión, ya que gestiona también el entorno del usuario. Incluso antes de ver el módulo en cuestión, el comando **ulimit** permite actuar en el entorno del shell y de los procesos que controla. El parámetro **-a** muestra las opciones controladas por **ulimit**:

```

$ ulimit -a
core file size      (blocks, -c) 0
data seg size       (kbytes, -d) unlimited

```

```

scheduling priority      (-e) 0
file size                (blocks, -f) unlimited
pending signals          (-i) 16380
max locked memory        (kbytes, -l) 32
max memory size        (kbytes, -m) 1753125
open files            (-n) 1024
pipe size                (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority        (-r) 0
stack size               (kbytes, -s) 8192
cpu time                  (seconds, -t) unlimited
max user processes      (-u) 16380
virtual memory           (kbytes, -v) 3333600
file locks                (-x) unlimited

```

Las líneas en negrita merecen su atención.

- ✦ **max memory size:** el tamaño de memoria máximo que puede ocupar el usuario;
- ✦ **open files:** el número máximo de descriptores de archivos; por lo tanto, el número máximo de archivos que se pueden abrir;
- ✦ **max user processes:** el número máximo de procesos que puede iniciar un usuario.

Se pueden cambiar estos valores según ciertos límites impuestos por el administrador. Existen límites soft (suaves o bajos) que son los valores por defecto devueltos por `ulimit`, y límites hard (duros, altos), que no se pueden superar.

Para pasar el número máximo de archivos abiertos a 2048:

```

$ ulimit -n 2048
$ ulimit -n
2048

```

El administrador root puede controlar los valores por defecto gracias al archivo `/etc/security/limits.conf`.

```
$ grep seb /etc/security/limits.conf
seb      hard  nproc   32768
seb      soft  nofile  1024
seb      hard  nofile  4096
```

En este ejemplo, el usuario Seb está limitado a un máximo de 32.768 procesos, puede abrir por defecto 1.024 archivos, pero puede montar 4.096 mediante una acción ulimit por su parte.

## 11. Los derechos SUDO

### a. Proporcionar privilegios extendidos

El comando **sudo** permite asignar el derecho a ejecutar comandos de administrador a uno o varios usuarios, en una o varias máquinas. En la práctica, para que un usuario pueda ejecutar un comando que, en principio, sólo puede ejecutar root, debe añadirse un derecho sudo a este usuario para este comando.

Los archivos de configuración de sudo son `/etc/sudoers` y los archivos en `/etc/sudoers.d/*`. Es posible editarlos a mano o con el comando **visudo**. Este último comando comprueba la sintaxis del archivo en el momento de guardar.

La clásica sintaxis de una línea sudo es la siguiente:

```
user  máquina = (user2) comando
```

- ~ **user**: el usuario (o alias) al cual se aplica la regla.
- ~ **máquina**: la máquina (o el alias) en la cual se aplica la regla.
- ~ **user2**: la cuenta con la cual el usuario ejecutará el comando.
- ~ **comando**: el comando que se va a ejecutar.

Por ejemplo, la línea siguiente va a autorizar al usuario Seb a ejecutar el comando fsck y sus parámetros con los derechos root sobre cualquier máquina (donde esté presente esta

regla):

```
seb    ALL = /sbin/fsck
```

Para utilizar **fsck**, Seb debe usar el comando **sudo** como a continuación:

```
seb@slyserver:~/handbrake> sudo /sbin/fsck
seb's password:
fsck 1.41.1 (01-Sep-2008)
e2fsck 1.41.1 (01-Sep-2008)
...
```

Por defecto, se pide la contraseña del usuario Seb antes de proseguir. El usuario puede obtener la lista de sus derechos sudo:

```
seb@slyserver:~/handbrake> sudo -l
User seb may run the following commands on this host:
(root) /sbin/fsck
```

Por defecto, se autentica el usuario una primera vez; luego, no se vuelve a pedir su contraseña mientras continúe en la misma sesión (mientras no cierre su consola o su entorno) o durante la duración indicada por la opción **timestamp timeout** del archivo de configuración.

La ventaja de **sudo**, además de asignar derechos puntuales a un grupo de personas determinadas, es la trazabilidad. Se transmiten los mensajes de **sudo** a **syslog**, que los puede volver a dirigir a un archivo. Puede ser `/var/log/messages`:

```
May 8 14:39:08 slyserver sudo:    seb
: TTY=pts/3 ; PWD=/home/seb ; USER=root ; COMMAND=/sbin/fsck
May 8 14:40:14 slyserver sudo:    seb
: TTY=pts/3 ; PWD=/home/seb ; USER=root ; COMMAND=list
```

Se puede establecer el registro de destino mediante `syslog.conf`, `rsyslog.conf` o `syslog-ng.conf`, según su distribución.



## b. Sintaxis de /etc/sudoers

El ejemplo anterior es deliberadamente limitado. Es posible:

- ˘ crear grupos de usuarios,
- ˘ crear grupos de máquinas,
- ˘ crear grupos de comandos,
- ˘ forzar o no el uso de una contraseña,
- ˘ forzar la ejecución de un comando bajo un usuario distinto de root.

Los grupos se llaman alias.

Para crear alias de usuarios, utilice esta sintaxis:

```
User_Alias ADMINS = seb, esteban, enrique
```

Si todos los administradores deben poder utilizar el comando **fsck**, la línea se convierte en:

```
ADMINS ALL= /sbin/fsck
```

Si los administradores sólo pueden ejecutar estos comandos en determinadas máquinas, cree un alias de máquinas:

```
Host_Alias SERVERS= slyserver, eeepc
```

La línea sudo se convierte en:

```
ADMINS SERVERS= /sbin/fsck
```

Se pueden añadir varios comandos sucesivamente, con o sin parámetros. Si los ADMINS deben ejecutar también el comando **/sbin/dumpe2fs**, la línea se convierte en esto:

```
ADMINS SERVERS=/sbin/fsck, /sbin/dumpe2fs
```

o, para saltar a la línea siguiente, ponga una barra oblicua "\" en el final de línea:

```
ADMINS SERVERS=/sbin/fsck, \  
/sbin/dumpe2fs
```

Puede crear alias de comandos para agruparlos:

```
Cmnd_Alias ADMCMD=/sbin/fsck, /sbin/dume2fs
```

La línea sudo se convierte en:

```
ADMINS SERVERS=ADMCMD
```

Para evitar la introducción de la contraseña de un usuario, escriba NOPASSWD como a continuación:

```
ADMINS SERVERS=NOPASSWD: ADMCMD
```

Los usuarios del alias ADMINS ya no tendrán que teclear contraseña para introducir los comandos del alias ADMCMD.

También puede forzar la utilización de una contraseña con PASSWD. Añada así el comando **mkfs**:

```
ADMINS SERVERS=NOPASSWD: ADMCMD, PASSWD:/sbin/mkfs
```

Para permitir que los ADMINS inicien un comando con la identidad de otro usuario distinto de root, sitúe el nombre del usuario (o alias) entre paréntesis, como a continuación:

```
ADMINS ALL=(esteban) PASSWD: /sbin/service
```

ADMINS podrá ejecutar /sbin/service como Esteban introduciendo la contraseña de éste.

El resultado del `sudo -l` asociado es el siguiente:

```
seb@slyserver:~> sudo -l
User seb may run the following commands on this host:
    (root) NOPASSWD: /sbin/fsck, /sbin/dumpe2fs
    (root) /sbin/mkfs
    (esteban) /sbin/service
```

Para iniciar el comando bajo otro usuario, utilice el parámetro `-u` de sudo:

```
seb@slyserver:~> sudo -u esteban /sbin/service
```

Puede precisar varios usuarios, pero también crear alias al igual que para el resto:

```
Runas_Alias LISTUSR=seb,esteban
```

Puede utilizar este alias en la línea sudo:

```
ADMINS ALL=(LISTUSR) PASSWD: /sbin/service
```

Puede utilizar el alias ALL: define el conjunto de usuarios, máquinas y comandos, según la posición donde está escrito. La línea siguiente significa que los ADMINS tienen permiso para ejecutar todos los comandos, en todas las máquinas, sin contraseña:

```
ADMINS ALL=NOPASSWD: ALL
```

El signo de exclamación permite excluir a determinados usuarios, comandos, máquinas, de un alias o de una lista. La línea siguiente da todos los derechos sobre todos los

programas sin contraseña a los ADMINS, salvo sobre `/sbin/mkfs`, que requiere una contraseña, y `/sbin/resize2fs`, que está prohibido:

```
ADMINS SERVERS=NOPASSWD: ALL, !/sbin/resize2fs, PASSWD:/sbin/mkfs
```

Si un ADMINS intenta iniciar `resize2fs`, obtiene un error:

```
seb@slyserver:~> sudo /sbin/resize2fs
Sorry, user seb is not allowed to execute '/sbin/resize2fs' as root
on slyserver.
```

La regla siguiente, muy peligrosa, autoriza a todo el mundo a ejecutar lo que quiera, en cualquier sitio y sin contraseña, como cualquier usuario:

```
ALL ALL=(ALL) NOPASSWD: ALL
```

A continuación viene una línea que se utiliza con frecuencia para autorizar a un usuario a hacerlo todo, salvo iniciar shells o el comando **su**, con la condición de que los alias SU y SHELLS estén completos:

```
seb ALL = ALL, !SU, !SHELLS
```

Esta línea no garantiza que Seb pueda obtener derechos por medios indirectos, por ejemplo, si se han vuelto a nombrar todos los comandos, o por rebotes de comandos sucesivos.

## 12. Auditoría más completa

Para efectuar la auditoría de un sistema, puede utilizar, además del acceso a los rastros e historiales, y de los comandos ya listados, los productos libres o gratuitos como tripwire, que comprueba la integridad del sistema, Nessus que vigila la seguridad del sistema, y

Crack, que detecta las contraseñas erróneas o la capa aplicativa de tipo ELK (*ElasticSearch Logstash Kibana*) para recolectar, indexar y buscar registros.

## 13. Los boletines de seguridad

### a. CERT: Computer Emergency Response Team

#### Historia

El primer virus capaz de replicarse en Internet (se llamaba Arpanet en aquella época) fue obra de un estudiante de la universidad de Cornell. Desarrollado por Robert Tappan Morris y lanzado a finales de 1988 sin intención de perjudicar, este programa se propagaba y replicaba sólo aprovechando fallos de seguridad de Unix y sus servicios. Este programa saturó rápidamente la red y las máquinas que había alcanzado, en torno a 6000, paralizando la red, compuesta entonces por 60.000 ordenadores, durante varios días. Sólo se había afectado al 10 % de las máquinas.

Para erradicar este virus se necesitó tiempo y muchos medios aportados por el MIT, Berkeley, etc. Hubo que estudiar su funcionamiento para entender cómo se había comportado y sólo entonces se pudieron corregir los agujeros de seguridad de los sistemas y servidores. Se aplicaron parches correctivos. El DARPA, iniciador del proyecto Arpanet (luego Internet) introdujo más tarde una nueva estructura llamada el CERT (CERT/CC, *CERT Coordination Center*), encargada de analizar las amenazas futuras y la vulnerabilidad de los sistemas.

Internet representa millones de máquinas interconectadas, con sistemas operativos y diferentes servicios. El CERT (<http://www.cert.org/>), cuya sede se encuentra a día de hoy en la universidad de Canergie Mellon, sigue estudiando las posibles vulnerabilidades en Internet, incluso a largo plazo, para obtener la mejor seguridad posible.

#### Papel del CERT

Las funciones del CERT son las siguientes:

- centralización y análisis de las peticiones de asistencia después de los incidentes de seguridad (ataques) sobre las redes y los sistemas de información: recepción de las peticiones, análisis de los síntomas y correlación de los

incidentes;

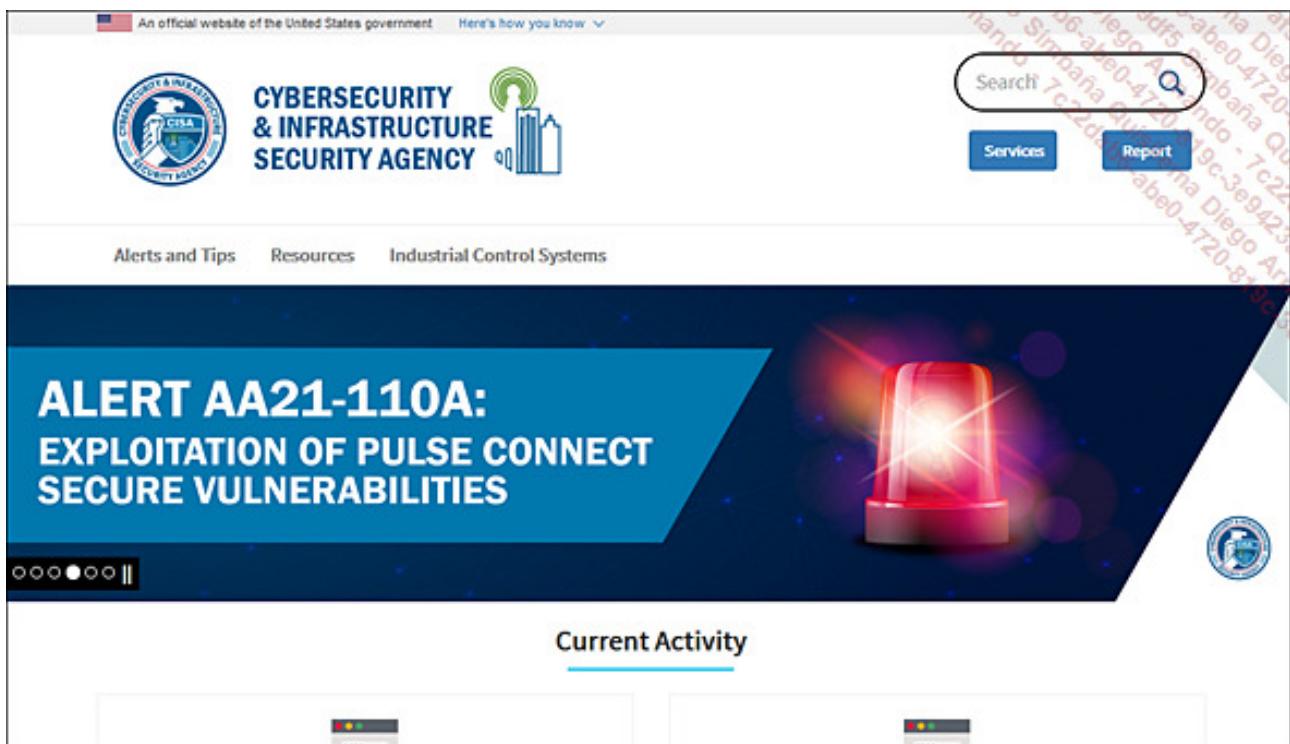
- ✓ tratamiento de las alertas y reacción a los ataques informáticos: análisis técnico, intercambio de información con otros CERT, contribución a estudios técnicos específicos;
- ✓ establecimiento y mantenimiento de una base de datos de las vulnerabilidades;
- ✓ prevención gracias a la publicación de información sobre las precauciones que es preciso tomar para minimizar los riesgos de incidente, o sus consecuencias;
- ✓ posible coordinación con otras entidades: centros de competencia de redes, operadores y proveedores de acceso a Internet, CERT nacionales e internacionales.

### Boletines del CERT

Existen varios tipos de comités CERT: por país, por industria, etc. Se emiten boletines independientes, pero vinculados entre ellos. A continuación indicamos dos sitios donde encontrar alertas:

<https://www.ccn-cert.cni.es>

Y sobre todo, <https://us-cert.cisa.gov/>



Alert AA21-110ª en la sede del CERT

Un ejemplo de alerta es un fallo de seguridad de la librería OpenSSL en Debian *Debian/Ubuntu OpenSSL Random Number Generator Vulnerability*, Referencia TA08-137A. Las claves aleatorias generadas desde la versión OpenSSL de Debian y Ubuntu no lo son realmente, lo que limita enormemente la seguridad de estas claves y conlleva un problema, ya que la actualización de la librería no es suficiente; también hay que generar de nuevo las claves existentes... Un ejemplo muy reciente es el fallo de seguridad heartbleed, que atañe a OpenSSL, referencia TA14-098A o el fallo Dirty COW del núcleo Linux, referencia CVE-2016-5195. Un último ejemplo es la alerta TA18-141A que trata los fallos materiales de Spectre y Meltdown, asociado al cinco CVE.

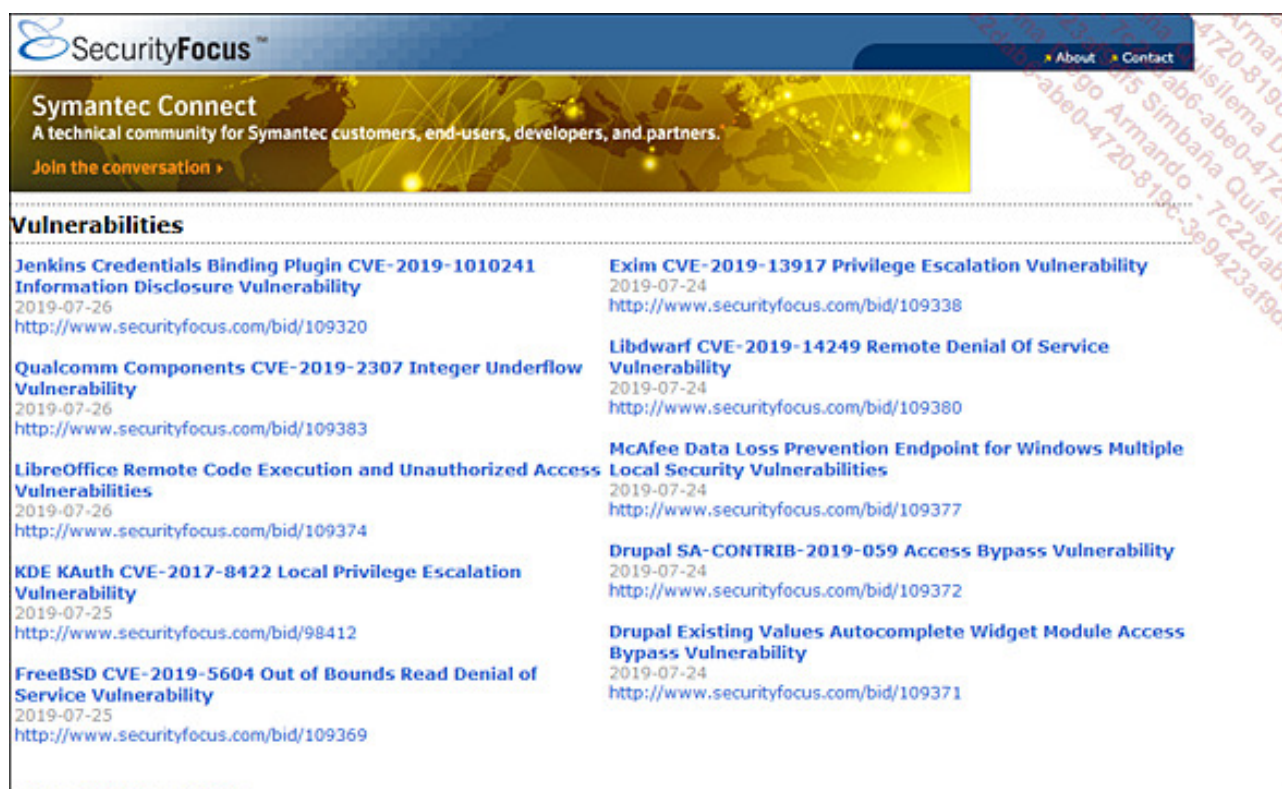
## b. SecurityFocus

**SecurityFocus** (anteriormente Bugtraq) es una lista de difusión electrónica (e-mailing) en línea desde 1993 que agrupa discusiones sobre vulnerabilidades, seguridad, anuncios y medios para detectar fallos y cómo corregirlos. Originalmente la lista nació de la necesidad de erradicar dos problemas:

- ˆ los muchos fracasos del CERT a la hora de prevenir problemas;
- ˆ la desidia de muchos editores y fabricantes, que no suministraban las actualizaciones de seguridad a pesar de los fallos encontrados.

Bugtraq pertenecía a SecurityFocus, que a su vez pertenece hoy al fabricante de las suites de seguridad de Symantec.

Puede darse de alta en la lista de distribución en el sitio <https://www.securityfocus.com/archive>



**SecurityFocus™**  
 Symantec Connect  
 A technical community for Symantec customers, end-users, developers, and partners.  
 Join the conversation »

**Vulnerabilities**

<b>Jenkins Credentials Binding Plugin CVE-2019-1010241 Information Disclosure Vulnerability</b> 2019-07-26 <a href="http://www.securityfocus.com/bid/109320">http://www.securityfocus.com/bid/109320</a>	<b>Exim CVE-2019-13917 Privilege Escalation Vulnerability</b> 2019-07-24 <a href="http://www.securityfocus.com/bid/109338">http://www.securityfocus.com/bid/109338</a>
<b>Qualcomm Components CVE-2019-2307 Integer Underflow Vulnerability</b> 2019-07-26 <a href="http://www.securityfocus.com/bid/109383">http://www.securityfocus.com/bid/109383</a>	<b>Libdwarf CVE-2019-14249 Remote Denial Of Service Vulnerability</b> 2019-07-24 <a href="http://www.securityfocus.com/bid/109380">http://www.securityfocus.com/bid/109380</a>
<b>LibreOffice Remote Code Execution and Unauthorized Access Vulnerabilities</b> 2019-07-26 <a href="http://www.securityfocus.com/bid/109374">http://www.securityfocus.com/bid/109374</a>	<b>McAfee Data Loss Prevention Endpoint for Windows Multiple Local Security Vulnerabilities</b> 2019-07-24 <a href="http://www.securityfocus.com/bid/109377">http://www.securityfocus.com/bid/109377</a>
<b>KDE KAuth CVE-2017-8422 Local Privilege Escalation Vulnerability</b> 2019-07-25 <a href="http://www.securityfocus.com/bid/98412">http://www.securityfocus.com/bid/98412</a>	<b>Drupal SA-CONTRIB-2019-059 Access Bypass Vulnerability</b> 2019-07-24 <a href="http://www.securityfocus.com/bid/109372">http://www.securityfocus.com/bid/109372</a>
<b>FreeBSD CVE-2019-5604 Out of Bounds Read Denial of Service Vulnerability</b> 2019-07-25 <a href="http://www.securityfocus.com/bid/109369">http://www.securityfocus.com/bid/109369</a>	<b>Drupal Existing Values Autocomplete Widget Module Access Bypass Vulnerability</b> 2019-07-24 <a href="http://www.securityfocus.com/bid/109371">http://www.securityfocus.com/bid/109371</a>

Página de inicio de SecurityFocus

### c. Los boletines de las distribuciones

Los editores de las mayores distribuciones facilitan también boletines de seguridad. Reutilizan a su vez las alertas de seguridad de otros organismos como el CERT o las listas de Bugtraq, pero, dado que cada editor es responsable de los paquetes que publica, debe publicar él mismo la correspondiente corrección y es habitual que parchee algunos productos y éstos difieran del original. De ahí la emisión de una alerta para prevenir a sus clientes y usuarios. A continuación mostramos los sitios donde puede obtener información de seguridad para las principales distribuciones:

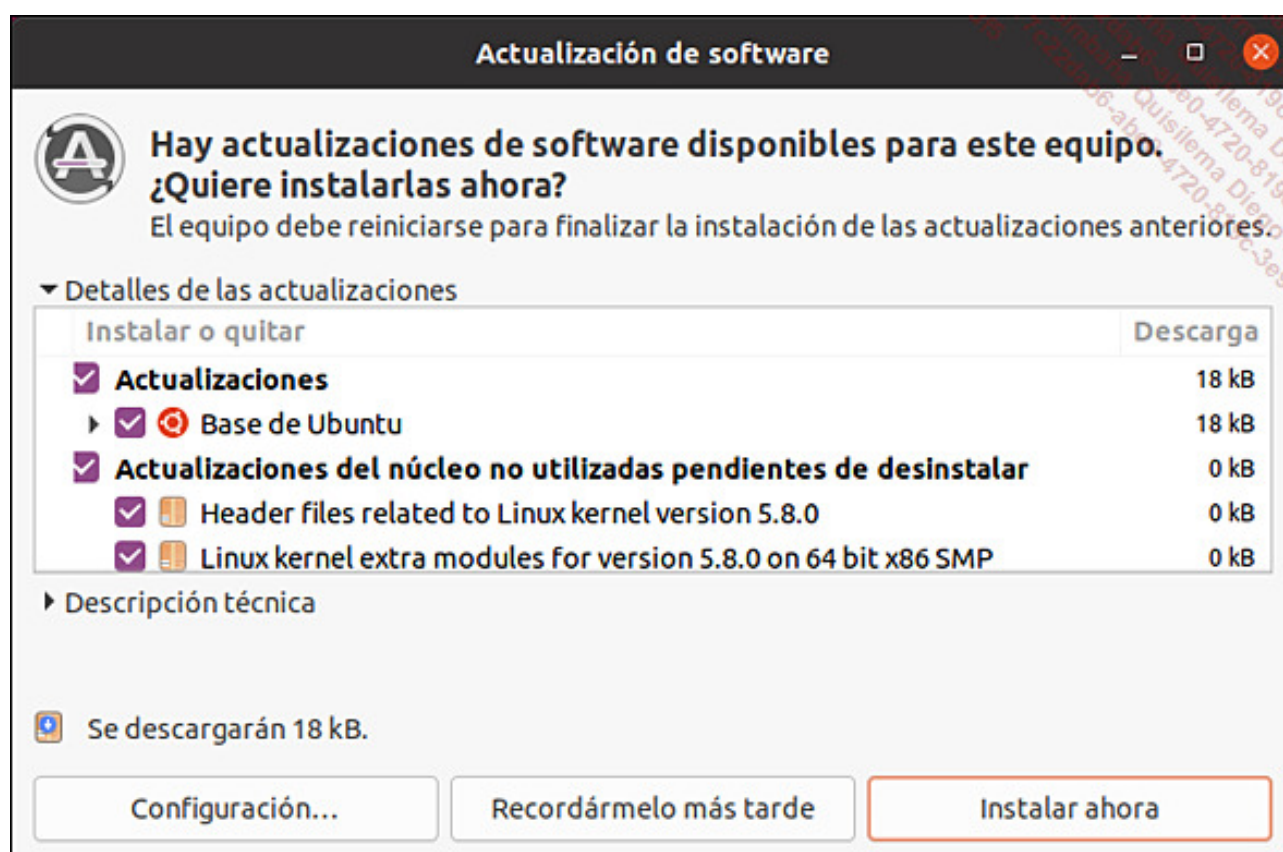
- ✓ Debian: <https://www.debian.org/security/>
- ✓ OpenSUSE: <https://lists.opensuse.org/opensuse-security-announce/>
- ✓ Fedora: <https://lists.fedoraproject.org/admin/lists/security.lists.fedoraproject.org/>
- ✓ Ubuntu: <https://www.ubuntu.com/usn>
- ✓ Red Hat Enterprise: <https://access.redhat.com/security/>



#### d. Los parches correctores

No basta con dar una alerta cuando se detecta un agujero de seguridad; también hace falta arreglarlo. Para ello, los editores suministran o bien paquetes corregidos (el vínculo se encuentra a menudo en el boletín de alerta), o bien parches correctores. Las distribuciones suministran habitualmente un componente que permite recuperar esos parches e informarle sobre su disponibilidad.

Cada distribución dispone de una herramienta que permite comprobar e informar al usuario de las actualizaciones del sistema. Esta herramienta puede ser distinta según la distribución; puede consultar la documentación proporcionada por el autor de la distribución para conocerla. Sin embargo, el uso de PackageKit (<https://www.freedesktop.org/software/PackageKit/>) es el más común, por lo menos su módulo de actualización. Es el caso de Ubuntu, OpenSUSE (y los productos Novell), Fedora y Red Hat en general. Iniciado en su interfaz gráfica, le informa desde la barra de tareas o la zona de notificación de las actualizaciones disponibles para su entorno. El siguiente ejemplo muestra la ejecución de la herramienta de actualización de un Ubuntu, llamada Update Manager que usa PackageKit:



*Update Manager de Ubuntu para las actualizaciones de seguridad*