educación IT
CODE YOUR FUTURE

Introducción a Linux

Módulo 4

Monitoreo y privilegios

Auditoría de archivos

Otra tarea importante es la de **chequear aquellos archivos que contengan permisos especiales *SUID*, *SGID* y *sticky bit***.

Estos tipos de permisos pueden llevar a que se ejecuten programas a los que algunos usuarios o grupos no deberían tener acceso.

Buscar archivos con SUID activo:

```
# find / -type f -perm +4000 2>/dev/null
/usr/sbin/pppd
/usr/bin/X
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/lppasswd
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/fping
/usr/bin/chsh
/usr/bin/sudo
(...salida cortada...)
```

Es lógico que algunos comandos figuren con este bit activado, ya que facilitan la administración de ciertas tareas, como las del comando passwd, que permite que cada usuario cambie su contraseña.

Para buscarlo de otra forma:

```
# find / -type f -perm -u=s -ls 2>/dev/null
7012450  992 -rwsr-xr-x   1 root    root        1011444 Aug 16  2013 /usr/sbin/vmware-authd
7012372  300 -rwsr-xr--   1 root    dip          302176 Jun 22  2012 /usr/sbin/pppd
9179027   12 -rwsr-sr-x   1 root    root         9508 May 11  2013 /usr/bin/X
9175219   72 -rwsr-xr-x   1 root    root        66196 May 25  2012 /usr/bin/gpasswd
9175223   44 -rwsr-xr-x   1 root    root        44564 May 25  2012 /usr/bin/chfn
9180127   16 -rwsr-xr-x   1 root    lpadmin     13712 Sep 29  2013 /usr/bin/lppasswd
9175220   48 -rwsr-xr-x   1 root    root        45396 May 25  2012 /usr/bin/passwd
```

Sgid

Ahora vamos a corroborar aquellos archivos que estén afectados por el *SGID*:

```
# find / -type f -perm -g=s -ls 2>/dev/null
6922941  132 -rwxr-sr-x   1 root    ssh         128396 Apr  2  2014 /usr/bin/ssh-agent
9175960   20 -rwxr-sr-x   1 root    tty          18020 Dec  9  2012 /usr/bin/wall
9177095   36 -rwxr-sr-x   1 root    crontab      34760 Jul  3  2012 /usr/bin/crontab
6923211   48 -rwsr-sr-x   1 daemon   daemon      46556 Jun  9  2012 /usr/bin/at
6922446  408 -rwxr-sr-x   1 root    utmp        410688 Sep 16  2012 /usr/bin/screen
9175222   56 -rwxr-sr-x   1 root    shadow       49364 May 25  2012 /usr/bin/chage
(...salida cortada...)
```

Otro ejemplo:

```
# find / -type f -perm +2000 2>/dev/null
/usr/bin/ssh-agent
/usr/bin/wall
/usr/bin/crontab
/usr/bin/at
/usr/bin/screen
/usr/bin/chage
(...salida cortada...)
```

Si quisiéramos buscar por ambos:

```
# find / -type f -perm +6000 2>/dev/null
/usr/sbin/pppd
/usr/bin/X
/usr/bin/gpasswd
/usr/bin/ssh-agent
/usr/bin/chfn
/usr/bin/wall
/usr/bin/passwd
(...salida cortada...)
```

Comando *fuser*

Se utiliza para **identificar procesos utilizando archivos o *sockets***.

Sintaxis:

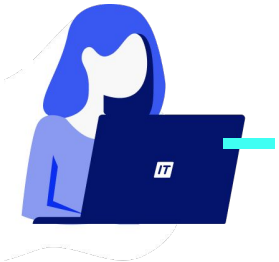
```
# fuser [opciones] archivo
```

Opciones	Descripción
-k	Envía `SIGKILL` al proceso que está accediendo el archivo definido.
-u	Muestra el nombre de usuario asociado al proceso.
-v	Modo verboso.

Ejemplo

Mostrar procesos y usuario asociados con un archivo:

```
# fuser -v /usr/sbin/apache2
                     USER         PID ACCESS COMMAND
/usr/sbin/apache2:   root         135 ...e. apache2
                     www-data     137 ...e. apache2
                     www-data     138 ...e. apache2
```



La letra que se ve a continuación del PID es el **tipo de acceso que se tiene al archivo**.

Las **letras más comunes** son:

- **e**: el ejecutable está corriendo.
- **F**: el archivo está abierto en modo escritura (solo en modo verboso).
- **f**: el archivo está abierto.

Comando *w*

El comando **w** muestra **la cantidad de usuarios conectados, el tipo que lleva iniciado el sistema y la carga promedio**.

```
# w
23:17:08 up  6:33,  3 users,  load average: 0,00,  0,02,  0,05
USER  TTY   LOGIN@       IDLE JCPU  PCPU  WHAT
root  tty2  16:48  4.00s      8.69s 0.09s w
root  tty3  18:43  4:33m      0.41s 0.41s -bash
```

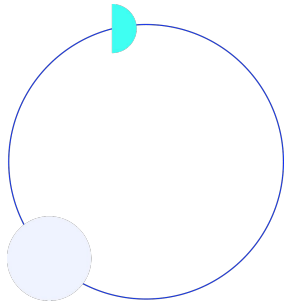
Veamos las opciones y su detalle en el siguiente slide.

Opciones	Descripción
LOGIN	Hora de conexión.
IDLE	Tiempo ocioso.
JCPU	Tiempo de uso que llevan todos los procesos en la `tty` definida.
PCPU	Tiempo usado por el proceso actual mostrado en la columna `WHAT`.

Comando *who*

El comando **who** muestra **quién se encuentra conectado en el sistema**.

```
# who
root   tty2   2014-12-11 16:44
root   tty3   2014-12-11 12:14
```



educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!