Universidad del Azuay

Facultad de Administración

Escuela de Ingeniería de Sistemas

Tema:
“Seguridades en Servidores Linux (Centos 4.2)”

Trabajo de graduación previo a la obtención del título de Ingeniero de Sistemas

Autores: Olga María Macancela León
Silvia Patricia Ramírez Procel

Director: Ing. Fernando Balarezo

Cuenca, Ecuador
2006

AGRADECIMIENTO

Dejo constancia de mi sincero agradecimiento a la Universidad del Azuay por haberme acogido y brindado su apoyo para poder lograr mi objetivo. A todos los maestros que con su enseñanza, me condujeron por el noble camino del saber.

Con el más sincero respeto, mi reconocimiento al Ing. Fernando Balarezo quien con sus amplios conocimientos contribuyó a la realización y culminación de la presente monografía.

A mi esposo, familiares y amigos quienes me brindaron su apoyo para poder culminar la carrera universitaria, y así poder cumplir una meta propuesta desde hace mucho tiempo.

Silvia Patricia Ramírez Procel

DEDICATORIA

Con mucho amor dedico esta monografía a mi esposo que siempre estuvo apoyándome en todo momento, hasta llegar a ser una profesional; a mis hijos que por ellos tuve la motivación de culminar lo propuesto.

A mis padres, hermanas(os) quienes me apoyaron a cada instante a lo largo de mi carrera, además de las innumerables muestras de cariño que me empujaron a seguir adelante y perseguir mis objetivos.

A mis amigos, amigas y compañeras que constituyeron para mi un apoyo fundamental, sin el cual no hubiera logrado muchas cosas. Me queda muy claro que en cada instante de la vida necesitamos de personas en quien depositar nuestra confianza, a pesar de que son ajenas a la familia, sin ningún reparo les entregamos toda nuestra amistad y cariño y desde luego esas personas saben retribuirlo.

Por éstas y otras razones dedico mi trabajo de investigación a quienes de alguna manera despertaron e incentivaron en mí el propósito de terminar lo que se empieza sin dejar nada inconcluso como es mi carrera universitaria.

Silvia Patricia Ramírez Procel

AGRADECIMIENTO

Al padre todopoderoso expreso mis infinitas gracias porque siempre me ha guiado por el camino del conocimiento y la sabiduría.

Mi sincera gratitud y reconocimiento al Ing. Fernando Balarezo, Director de la Tesis, quien siempre estuvo atendiendo mis peticiones contribuyendo de esta manera a la realización de la monografía. También quiero hacer extensivo mi agradecimiento al Ing. Ernesto Pérez quien siempre estuvo con apertura a mis inquietudes.

A mis familiares y amigos quienes me supieron brindar su apoyo incondicional y estuvieron en los momentos que más los necesité, pudiendo de esta manera brindarme seguridad en el camino que estaba tomando.

Olga María Macancela León

DEDICATORIA

Dedico este trabajo investigativo a mi padre y hermanos, quienes hoy van a ver cristalizados sus anhelos de ser una profesional, y pueda cumplir mis sueños.

A mis amigos, amigas y compañeras que con el sentimiento de compañerismo supieron brindarme su apoyo en momentos que lo necesitaba, siendo una señal para que me diera cuenta que verdaderos amigos que te abran el corazón son difíciles de encontrar y soy afortunada en tenerlos a mi lado.

Olga María Macancela León

INDICES DE CONTENIDO

RESUMEN

La presente monografía dio un enfoque breve de los principales puntos y herramientas que deben tomarse en cuenta para la Seguridad de un Sistema Linux, lo cual nos permitirá definir ciertos criterios aplicando nuestro sentido común para responder estas interrogantes que se nos presentaran dependiendo de nuestra situación como pueden ser:

- ¿Qué queremos proteger?
- ¿Qué valor tiene lo que queremos proteger?
- ¿Qué costo tiene la seguridad?
- ¿De quién nos queremos proteger?
- ¿Cuáles son los puntos débiles de nuestro sistema?

Las posibles respuestas a estas preguntas nos proporcionan muchas posibilidades como para poderlo tratar todo.

Lo primero que tenemos que determinar es lo que queremos proteger. No será lo mismo una estación de trabajo personal aislada con conexiones a Internet esporádicas que un servidor web con conexión permanente o un cortafuegos.

También tendremos que considerar el costo de lo que queremos proteger: posible costo económico, tiempo de restauración o instalación, prestigio, perdida de clientes, etc. También el costo de la seguridad en unos términos parecidos a los anteriores. Sería absurdo que invirtiéramos más en protección que el costo de lo protegido.

También hay que considerar que existe una relación inversa entre seguridad y funcionalidad. Cuanto más seguro hacemos un sistema, menos funcional resulta, ofreciendo menos servicios y más limitaciones de acceso. Esto también constituye otro costo adicional: facilidad de uso.

Después de saber qué y de qué tenemos que protegernos, de quiénes y cuáles son sus posibles objetivos, y viendo los servicios que necesariamente hay que prestar o usar, obtendremos un esquema elemental de nuestra situación y de las medidas que tenemos que tomar.

ABSTRACT

These monograph gave a brief focus of the main points and tools which should be taken into account to the Security of a Linux System, which will allow us to define certain criteria applying our common sense; in order to answer some questions, which may present. It depends on our situation, those questions could be the following:

- What do we want to protect ?
- What value does what we want to protect have ?
- What cost does the security have ?
- Who do we want to protect from ?
- What are the weak parts of our system ?

The possible answers to these questions provide us many possibilities to be able to treat it. The first thing we have to determine is what we want to protect. An isolated personal working station with sporadic connections to Internet won't be the same as a web server with permanent connection or a firewall.

Also, we'll have to take into consideration the cost of what we want to protect: the possible financial cost, the restoration or installation time, the prestige, the loss of clients, etc. In addition, the security cost in similar terms to the previous ones. It would be an absurd to invest more money in protection than in the cost of what is being protected.

It is also important to consider that there is an inverse relationship between security and functionality. The more reliable we make a system, the less functional it is; so, it offers less services and more access limitations. It also represents an additional cost: use facility.

After we know what and from what we have to protect ourselves, from whom, and which their possible objectives are. Therefore, looking at the services which necessarily we have

to give or to use, we’ll get an elementary outline of our situation and of the steps we have to follow.

I. INTRODUCCION

La seguridad se está convirtiendo en una cuestión cada vez más importante. Ahora, la seguridad es un requisito básico, ya que la red global es insegura por definición. Mientras sus datos estén almacenados en un soporte informático, mientras sus datos vayan desde un sistema X a otro sistema Y a través de un medio físico, Internet, por ejemplo, puede pasar por ciertos puntos durante el camino proporcionando a otros usuarios la posibilidad de interceptarlos, y alterar la información contenida. Incluso algún usuario de su sistema puede modificar datos de forma maliciosa para hacer algo que nos pueda resultar perjudicial. Con el acceso masivo y barato a Internet se han reducido notablemente los costes de un atacante para asaltar un sistema en red, a la vez que ha aumentado paralelamente el número de potenciales atacantes.

También queremos remarcar el carácter dinámico de la seguridad de los sistemas en red. Continuamente aparecen nuevos métodos para conseguir accesos indebidos o comprometer el correcto funcionamiento de la red. Esto obliga a estar actualizado permanentemente, y consultar las publicaciones electrónicas que informan de los últimos sucesos detectados.

Se pretende mostrar algunos métodos y herramientas generales para prevenir que los ataques con ciertos programas maliciosos tengan éxito en su sistema.

CAPITULO I

1. SEGURIDAD DEL SISTEMA DE ARCHIVOS

1.1 Introducción

Una norma básica de seguridad radica en la correcta asignación de los privilegios que le corresponden a cada usuario para poder cubrir las necesidades de su trabajo sin poner en riesgo el trabajo de los demás y evitando el uso indebido de los recursos.
Otra forma de mantener la correcta seguridad y funcionabilidad del sistema es conocer como esta distribuida la información en el árbol de directorios.

1.2 Directorios

El sistema de archivos bajo Linux es una estructura de árbol construida bajo archivos y directorios. Linux almacena distintos tipos de información sobre cada archivo en su sistema de archivos, incluyendo la siguiente información:

- El nombre del archivo.
- El tipo de archivo.
- El tamaño del archivo.
- La ubicación física del archivo en el disco.
- Varios horarios de permisos controlan los derechos de acceso del propietario, los miembros del grupo asociado y otros utilitarios. Si se utiliza la instrucción **ls -l** se creará una lista de archivos que muestra el campo de permisos. Los permisos del archivo se modifican mediante la instrucción chmod.

“Dos categorías de archivos pueden definirse en Linux: Los que se pueden y no compartir, y estáticos y no estáticos. Servirá para entender la estructura del árbol de directorios.

En una red es normal que existan documentos y aplicaciones que puedan compartirse para ahorrar espacio. Pero existe información relativa a un ordenador, que no debe compartirse. Los datos estáticos solo son modificables por el administrador de sistema o root.

En el directorio raíz contiene toda la estructura jerárquica del sistema.

/bin Este directorio contiene archivos binarios esenciales del sistema, disponibles para todos los usuarios. Por lo que es estático y no se puede compartir. Normalmente no incluye carpetas.

/boot Los archivos necesarios para el arranque de Linux se encuentran en esta carpeta. Estático y no puede compartirse.

/dev Archivos de dispositivos del sistema. Linux trata todo como si fuera un archivo incluido el Hardware. La disquetera es el fichero /dev/fdo. Si el disco duro está conectado como maestro al IDE primario es el fichero hda y hdd si es esclavo del secundario, etc. Estático y no puede compartirse.

/etc Aquí están los archivos de configuración locales, o sea, los específicos del ordenador donde corre el sistema. Estático y no puede compartirse. Algunos de los ficheros que contiene:

passwd Usuarios del sistema y sus contraseñas

inittab Configuración de init

bashrc Configuración de shell

/home Aquí están los directorios personales del sistema y los usuarios. No es estático pudiéndose o no compartirse.

/lib Dentro están las librerías compartidas, se necesitan para ejecutar las instrucciones y arrancar el sistema. También los módulos esenciales del kernel. Estático, no compartible.

/mnt Se encuentran los puntos de montaje de los sistemas de archivos temporales.

/opt Archivos de paquetes y aplicaciones.

/proa Sistema de archivo virtual. (procesos de la memoria y el kernel).

/root Dentro está el directorio personal del administrador del sistema. No estático, incompartible.

/sbin Archivos binarios necesarios para arrancar el sistema. Estático, no compartible.

/tmp Archivos temporales de aplicaciones. No estático, incompartible.

/usr Directorio principal de almacenamiento de aplicaciones. Se encuentra en una partición diferente al resto del sistema. Dispone de su propia estructura jerárquica.

/usr/X11R6. Para X Window.

/usr/bin. Contiene los ejecutables no necesarios para la administración y mantenimiento del sistema.

/usr/sbin. Librerías necesarias para los binarios.

/usr/local. Archivos de las aplicaciones.

/var Contiene archivos que pueden modificarse durante el uso del sistema pero que deben quedar como estaban al volver a arrancar Linux. "[1]

Un primer criterio para mantener un sistema seguro sería hacer una correcta distribución del espacio de almacenamiento. Esto limita el riesgo de que el deterioro de una partición afecte a todo el sistema. La pérdida se limitaría al contenido de esa partición. No hay unas normas generales aplicables; el uso al que vaya destinado el sistema y la experiencia son las bases de la decisión adecuada, aunque sí podemos dar algún consejo:

- Si el sistema va a dar servicio a múltiples usuarios que requieren almacenamiento para sus datos privados, sería conveniente que el directorio /home tuviera su propia partición.
- Si el equipo va a ser un servidor de correo, impresión, etc., el directorio /var o incluso /var/spool podrían tener su propia partición.

[1] http://quicksitebuilder.cnet.com/camiloalie/ElRinconDeWindows/id124.html

- Algunos directorios son necesarios en la partición raíz. Contienen datos que son necesarios durante el proceso de arranque del sistema. Son /dev/, /etc, /bin, /sbin, /lib, /boot.
- El directorio /usr/local contiene los programas compilados e instalados por el administrador. Resulta conveniente usar una partición propia para proteger estos programas personalizados de futuras actualizaciones del sistema. Este criterio también se puede aplicar al directorio /opt.

1.3 Permisos

Tenemos que asegurarnos de que los ficheros del sistema y los de cada usuario sólo son accesibles por quienes tienen que hacerlo y de la forma que deben. No sólo hay que protegerse de ataques o miradas indiscretas, también hay que protegerse de acciones accidentales.

"En general, cualquier sistema UNIX divide el control de acceso a ficheros y directorios en tres elementos: `propietario`, `grupo` y `otros`. Tanto el propietario como el grupo son únicos para cada fichero o directorio. Eso sí, a un grupo pueden pertenecer múltiples usuarios. `Otros` hacen referencia a los usuarios que ni son el propietario ni pertenecen al grupo.

A continuación presentamos una breve explicación a cerca de los permisos.

Propiedad:

Qué usuario y grupo posee el control de los permisos del i-nodo. Se almacenan como dos valores numéricos, el uid (user id) y gid (group id).

Permisos:

Bits individuales que definen el acceso a un fichero o directorio. Los permisos para directorio tienen un sentido diferente a los permisos para ficheros. Más abajo se explican algunas diferencias.

- **Lectura (r)**:
    - **Fichero**: Poder acceder a los contenidos de un fichero
    - **Directorio**: Poder leer un directorio, ver qué ficheros contiene
- **Escritura (w)**:
    - **Fichero**: Poder modificar o añadir contenido a un fichero
    - **Directorio**: Poder borrar o mover ficheros en un directorio
- **Ejecución(x)**:
    - **Fichero**: Poder ejecutar un programa binario o guión de shell
    - **Directorio**: Poder entrar en un directorio

Además tenemos otros bits de permisos que no podemos pasar por alto cuando estamos tratando de temas de seguridad.

Sticky bit:

El sticky bit tiene su significado propio cuando se aplica a directorios. Si el sticky bit está activo en un directorio, entonces un usuario sólo puede borrar ficheros que son de su propiedad o para los que tiene permiso explícito de escritura, incluso cuando tiene acceso de escritura al directorio. Esto está pensado para directorios como /tmp, que tienen permiso de escritura global, pero no es deseable permitir a cualquier usuario borrar los ficheros que quiera. El sticky bit aparece como '**t**' en los listados largos de directorios.

drwxrwxrwt 19 root root 8192 Jun 24 14:40 tmp

Attributo SUID: (Para Ficheros)

Este bit describe permisos al identificador de usuario del fichero. Cuando el modo de acceso de ID de usuario está activo en los permisos del propietario, y ese fichero es ejecutable, los procesos que lo ejecutan obtienen acceso a los recursos del sistema basados en el usuario que crea el proceso (no el usuario que lo lanza). Por ejemplo /usr/bin/passwd es un ejecutable

propiedad de root y con el bit SUID activo. ¿Por qué? Este programa lo puede usar cualquier usuario para modificar su clave de acceso, que se almacena en

-rw-r--r-- 1 root root 1265 Jun 22 17:35 /etc/passwd

pero según los permisos que observamos en este fichero, sólo root puede escribir y modificar en él. Entonces sería imposible que un usuario pudiera cambiar su clave si no puede modificar este fichero. La solución para este problema consiste en activar el bit SUID para este fichero:

-r-s--x--x 1 root root 10704 Apr 14 23:21 /usr/bin/passwd

de forma que cuando se ejecute, el proceso generado por él es un proceso propiedad de root con todos los privilegios que ello acarrea.

Atributo SGID: (Para ficheros)

Si está activo en los permisos de grupo, este bit controla el estado de "poner id de grupo" de un fichero. Actúa de la misma forma que SUID, salvo que afecta al grupo. El fichero tiene que ser ejecutable para que esto tenga algún efecto.

Atributo SGID: (Para directorios)

Si activa el bit SGID en un directorio (con "chmod g+s directorio"), los ficheros creados en ese directorio tendrán puesto su grupo como el grupo del directorio.

A continuación se describen los significados de los permisos de acceso individuales para un fichero. Normalmente un fichero tendrá una combinación de los siguientes permisos:

-r-------- Permite acceso de lectura al propietario
--w------- Permite modificar o borrar el fichero al propietario
---x------ Permite ejecutar este programa al propietario, (los guiones de shell también requieren permisos de lectura al propietario)
---s------ Se ejecutará con usuario efectivo ID = propietario
-------s-- Se ejecutará con usuario efectivo ID = grupo
-rw------T No actualiza "instante de última modificación". Normalmente usado para ficheros de intercambio (swap).
---t------ No tiene efecto. (antes sticky bit)

A continuación se describen los significados de los permisos de acceso individuales para un directorio. Normalmente un directorio tendrá una combinación de los siguientes permisos:

dr-------- Permite listar el contenido pero no se pueden leer los atributos.
d--x------ Permite entrar en el directorio y usar en las rutas de ejecución completas.
dr-x------ Permite leer los atributos del fichero por el propietario.
d-wx------ Permite crear/borra ficheros.
d------x-t Previene el borrado de ficheros por otros con acceso de escritura. Usado en /tmp.
d---s--s-- No tiene efecto

Los ficheros de configuración del sistema (normalmente en /etc) es habitual que tengan el modo `640 (-rw-r-----)`, y que sean propiedad del root. Dependiendo de los requisitos de seguridad del sistema, esto se puede modificar. Nunca deje un fichero del sistema con permiso de escritura para un grupo o para otros. Algunos ficheros de configuración, incluyendo /etc/shadow, sólo deberían tener permiso de lectura por root, y los directorios de /etc no deberían ser accesibles, al menos por otros.

SUID Shell Scripts

Los scripts de shell SUID son un serio riesgo de seguridad, y por esta razón el núcleo no los acepta. Sin importar lo seguro que piense que es su script de shell, puede ser utilizado para que un cracker pueda obtener acceso a una shell de root." [2]

1.4 Integridad de los Archivos: Tripwire

"Los ataques exitosos a través de la red típicamente involucran la modificación parcial del sistema mediante la alteración o reemplazo de ciertos archivos, lo cual suele ser empleado por el atacante para posteriormente tomar el control total del sistema.

Tripwire asume que todos los controles de seguridad han fallado, y que nuestro sistema ya ha sido alterado; al menos, parcialmente. Sin embargo, parte del arte de los atacantes consiste en no ser descubiertos, y para esto emplean diversas técnicas relativamente sofisticadas. Tripwire servirá para alertar al administrador de estos cambios (los cuales de

[2] http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html

otro modo podrían pasar desapercibidos por semanas o meses) a fin de tomar acciones con rapidez.

Para esto, Tripwire monitorea rutinariamente la integridad de una gran cantidad de archivos que tienden a ser blanco de los atacantes. Sin embargo, este proceso es pesado, y se suele ejecutar a intervalos; por ejemplo, diarios o interdiarios, aunque no hay ninguna restricción (salvo de recursos) para no lanzarlo cada media hora."[3]

1.4.1 Instalar Tripwire

Descargue la versión open source de Tripwire del sitio:

http://centos.karan.org/el4/extras/stable/i386/RPMS/tripwire-2.3.1-21.i386.rpm

Ahora instálelo:

rpm –Uvh tripwire-2.3.1-21.i386.rpm

1.4.2 Configuración de Tripwire

1.4.2.1 Definir las claves de Tripwire

Tripwire utiliza dos claves (que pueden ser palabras u oraciones que deben contener máximo 8 caracteres que pueden ser una combinación de números y letras) para almacenar su información. Una de ellas, la "site key" o "clave del sitio", se emplea para encriptar los archivos de configuración y de las políticas. La otra - la "local key" o "clave local", se usa para encriptar la información referida al estado de los archivos del sistema que se monitorean.

Ud. necesita estas dos claves para las tareas de administración de Tripwire. Estas se deben introducir tan pronto como se ha instalado Tripwire mediante la instrucción:

/ tripwire-setup-keyfiles

Esta entrada generará lo siguiente:

[3] http://es.tldp.org/Tutoriales/GUIA_TRIPWIRE/guia_tripwire.html

---

Creating key files...

(When selecting a passphrase, keep in mind that good passphrases typically have upper and lower case letters, digits and punctuation marks, and are at least 8 characters in length.)

Enter the site keyfile passphrase:
Verify the site keyfile passphrase:
Generating key (this may take several minutes)...Key generation complete.

(When selecting a passphrase, keep in mind that good passphrases typically have upper and lower case letters, digits and punctuation marks, and are at least 8 characters in length.)

Enter the local keyfile passphrase:
Verify the local keyfile passphrase:
Generating key (this may take several minutes)...Key generation complete.

Signing configuration file...
Please enter your site passphrase:
Wrote configuration file: /etc/tripwire/tw.cfg

A clear-text version of the Tripwire configuration file:
/etc/tripwire/twcfg.txt
has been preserved for your inspection. It is recommended that you move this file to a secure location and/or encrypt it in place (using a tool such as GPG, for example) after you have examined it.

Signing policy file...
Please enter your site passphrase:

Wrote policy file: /etc/tripwire/tw.pol
A clear-text version of the Tripwire policy file:
/etc/tripwire/twpol.txt
has been preserved for your inspection. This implements a minimal policy, intended only to test essential Tripwire functionality. You should edit the policy file to describe your system, and then use
twadmin to generate a new signed copy of the Tripwire policy.

Once you have a satisfactory Tripwire policy file, you should move the clear-text version to a secure location and/or encrypt it in place (using a tool such as GPG, for example).

Now run "tripwire --init" to enter Database Initialization Mode. This reads the policy file, generates a database based on its contents, and then cryptographically signs the resulting database. Options can be entered on the command line to specify which policy, configuration, and key files are used to create the database. The filename for the database can be specified as well. If no options are specified, the
default values from the current configuration file are used.

1.4.2.2 Construir la base de datos Tripwire

Una vez inicializadas las claves, podemos proceder a initializar la BD de tripwire, digamos que con este paso el tripwire revisa el estado actual de nuestro sistema y lo guarda como una imagen buena de él. Es importante que el tripwire se inicialice sobre una máquina recién instalada y posiblemente no conectada a la internet, para saber que está perfectamente trabajando y sin problemas de ningún tipo.

tripwire --init

Please enter your local passphrase:

Parsing policy file: /etc/tripwire/tw.pol

Generating the database...

*** Processing Unix File System ***

Warning: File system error.

Filename: /dev/kmem

No such file or directory

Continuing...

Warning: File system error.

Filename: /proc/ksyms

No such file or directory

Continuing...

Warning: File system error.

Filename: /proc/scsi

No such file or directory

Continuing...

Warning: File system error.

Filename: /usr/sbin/fixrmtab

No such file or directory

Continuing...

Básicamente este paso revisará todos y cada uno de los archivos del sistema, sobre todo los binarios fundamentales de él. Una vez acabe tendremos en una BD generada por el tripwire, almacenados los datos de esta corrida inicial. Posteriores corridas del tripwire compararán todo el sistema contra esta BD para así detectar potenciales errores y violaciones de integridad.

1.4.2.3 Verificación del filesystem

Realizando chequeos

El tripwire instala en /etc/cron.daily un script que se ejecuta diariamente generando un reporte de cambios en el sistema.

Este reporte se genera en la madrugada, ahora si quisiéramos ejecutarlo manualmente (no es obligatorio) podríamos hacerlo con:

tripwire –m c

Demorará un rato y nos generará un reporte con la fecha actual. Repetimos, estos mismos reportes se generan diariamente en la madrugada por lo que no es necesario.

1.4.2.4 Revisando Reportes

Los reportes son almacenados cada vez que realizamos un chequeo en el directorio /var/lib/tripwire/report

Estos reportes están encriptados y no se pueden cambiar si no es con la clave del tripwire. Por lo que un potencial atacante no lo podría modificar para su conveniencia.
Si deseamos ver el contenido de cualquier reporte digite la siguiente instrucción:

twprint -m r --twrfile /var/lib/tripwire/report/localhost.localdomain-20060305-165902.twr |less

Reporte de ejemplo:

Note: Report is not encrypted.

Tripwire(R) 2.3.0 Integrity Check Report

Report generated by: root

Report created on: Tue 27 Sep 2005 09:44:25 AM ECT

Database last updated on: Never

```
===============================================================
Report Summary:
===============================================================

Host name:                localhost.localdomain

Host IP address:          127.0.0.1

Host ID:                  None

Policy file used:         /etc/tripwire/tw.pol

Configuration file used:  /etc/tripwire/tw.cfg

Database file used:       /var/lib/tripwire/localhost.localdomain.twd

Command line used:        /usr/sbin/tripwire --check

===============================================================
Rule Summary:
===============================================================
  Section: Unix File System
---------------------------------------------------------------------------

  Rule Name                     Severity Level    Added    Removed Modified
  ---------                     --------------    -----    ------- --------
  Invariant Directories           66              0        0       0

  Temporary directories           33              0        0       0

* Tripwire Data Files             100             1        0       0

 .......
 ........
 ........
 *** End of report ***

Tripwire 2.3 Portions copyright 2000 Tripwire, Inc. Tripwire is a registered
trademark of Tripwire, Inc. This software comes with ABSOLUTELY NO WARRANTY;

for details use --version. This is free software which may be redistributed

or modified only under certain conditions; see COPYING for details.
```

1.4.2.5 Configurar el archivo de políticas

La configuración de los archivos que van a ser monitoreados por Tripwire se mantiene en un gran archivo conocido como "archivo de políticas" (policy file.) Su manipulación es algo tediosa dada su extensión. Tripwire viene con un archivo que sirve de "plantilla" para ser modificado. Este archivo es: `/etc/tripwire/twpol.txt`.

Ud. puede modificarlo directamente con un editor de texto (aunque le aconsejo que guarde una copia sin modificar del mismo.)

Ahora haremos una observación de órden práctico y didáctico: Tripwire por lo general toma varios minutos en cada una de sus ejecuciones, y si Ud. nunca lo ha usado, probablemente le resultará desesperante aguardar mucho tiempo sin saber si las cosas están yendo bien o mal. Por este motivo yo sugiero que empecemos con una versión reducida (y casi inútil) del archivo de políticas. Una vez que Ud. comprenda el proceso completo, podrá retomar el archivo original y aprovecharlo.

Nuevamente va la advertencia: haga una copia de seguridad del archivo `twpol.txt`.

Para recortar el archivo proporcionado, simplemente use un editor de texto (como vi) y busque la sección "File System and Disk Administraton Programs" (que en el archivo que yo tengo, se ubica a partir de la línea 185.) Un extracto de esa sección es presentado aquí:

```
############################################### #
# File System and Disk Administration Programs # #
###############################################
(
  rulename = "File System and Disk Administraton Programs",
  severity = $(SIG_HI)
)
{
  /sbin/accton                    -> $(SEC_CRIT) ;
  /sbin/badblocks                   -> $(SEC_CRIT) ;
```

```
  /sbin/dosfsck                -> $(SEC_CRIT) ;
  /sbin/e2fsck                 -> $(SEC_CRIT) ;
  /sbin/debugfs                -> $(SEC_CRIT) ;
  /sbin/dumpe2fs               -> $(SEC_CRIT) ;
  /sbin/dump                   -> $(SEC_CRIT) ;
```

Como Ud. ya se imaginará, esto corresponde a un conjunto de archivos que se monitorean por Tripwire. Nosotros reduciremos la extensa lista recortando el archivo en esta sección. Por ejemplo, haciendo que termine en **/sbin/e2fsck**:

Como se ve, hemos recortado la parte que estaba más abajo de **/sbin/e2fsck**, y hemos tenido cuidado de añadir una llave de cierre (}) para mantener la sintaxis del archivo. A fin de ver los posibles errores con que nos podemos encontrar, sugiero al lector que añada un archivo inexistente a la lista:

Tenga cuidado de no insertar este archivo por debajo de la llave de cierre del grupo. Recuerde que más tarde deberá vérselas con el archivo original que contiene (en mi caso) 452 entradas.

```
...
################################################ #
# File System and Disk Administration Programs # #
################################################
(
  rulename = "File System and Disk Administraton Programs",
  severity = $(SIG_HI)
)
{
  /sbin/accton                 -> $(SEC_CRIT) ;
  /sbin/badblocks              -> $(SEC_CRIT) ;
  /sbin/dosfsck                -> $(SEC_CRIT) ;
  /sbin/e2fsck                 -> $(SEC_CRIT) ;
```

```
# ARCHIVO DE PRUEBA INEXISTENTE AÑADIDO. OBSERVE QUE ESTA
# UBICADO ANTES DE LA LLAVE DE CIERRE
  /sbin/lechuga                    -> $(SEC_CRIT) ;
}

# AHORA AQUI TERMINA EL ARCHIVO
```

1.4.2.6 Instalar el archivo de políticas

“Cuando el archivo de políticas contiene todo lo que pretendemos monitorear, es menester "instalarlo". En realidad Tripwire usa una versión compilada y encriptada de este archivo, que se almacena en `/etc/tripwire/tw.pol`. Para generarlo (y regenerarlo cuantas veces se necesite), usar:

```
# twadmin -m P /etc/tripwire/twpol.txt
```

1.4.2.7 Configuración permanente de Tripwire

Automatización

Ahora que Ud. ha probado la correcta ejecución de Tripwire, debemos programar su ejecución automática. Se aconseja una frecuencia diaria, aunque el administrador es libre de usar otro esquema. En RedHat 7.1, la ejecución diaria de tripwire se efectúa fácilmente creando un archivo en el directorio `/etc/cron.daily` (por ejemplo, `/etc/cron.daily/tripwire` con el siguiente contenido:

```
/usr/sbin/tripwire -m c | mail root@localhost
```

Donde Ud. deberá modificar la dirección "root@localhost" por lo que más le convenga. No olvide asegurarse de que el servicio **cron** esté operativo.

Asegúrese de que este archivo para cron sea ejecutable:

chmod 755 /etc/cron.daily/tripwire”[4]

1.5 Limitar el espacio asignado a los usuarios

“Un ataque posible a cualquier sistema es intentar consumir todo el espacio del disco duro. Una primera protección contra este ataque es separar el árbol de directorios en diversos discos y particiones. Pero esto puede no ser suficiente y por eso el núcleo del sistema proporciona la posibilidad de controlar el espacio de almacenamiento por usuario o grupo.

Lo primero que tendríamos que hacer es asegurarnos de que nuestro núcleo tiene soporte para las cuotas de usuarios.

```
# dmesg | grep quotas
VFS: Diskquotas version dquot_6.4.0 initialized
```

En caso contrario, el núcleo no ha sido compilado con soporte para el sistema de cuotas para usuarios. En este caso será necesario compilar un nuevo núcleo Linux.

Ahora es necesario editar el fichero /etc/fstab y añadir usrquota o grpquota en la partición o particiones en las que nos interese limitar el espacio de almacenamiento.

El siguiente ejemplo establece el sistema de cuotas para el uso del directorio /home montado en la partición /dev/hda3:

```
/dev/hda3 /home ext2 defaults,usrquota 1 2
```

Ahora podemos recopilar la información de las particiones donde se haya definido el sistema de cuotas. Podemos usar la instrucción:

```
/sbin/quotachek -av

Scanning /dev/hda3 [/home] done
Checked 215 directories and 2056 files
```

[4] http://es.tldp.org/Tutoriales/GUIA_TRIPWIRE/guia_tripwire.html

Using quotafile /home/quota.user
Updating in-core user quotas

Al ejecutar esta instrucción también se crea un fichero llamado quota.user o quota.grp en la partición o particiones afectada(s).

ls -la /home/quota.user

-rw------- 1 root root 16224 Feb 04 14:47 quota.user

Ya está activo el sistema de cuotas y la próxima vez que se inicie el sistema, se activarán automáticamente en todas las particiones que se hayan definido. Ya no será necesario volver a ejecutar manualmente esta intrucción, ya que el sistema lo hará de forma automática al comprobar y montar cada uno de los sistemas de ficheros desde el fichero /etc/rc.d/rc.sysinit. El siguiente paso es definir la cuotas para cada usuario. Para ello existen dos métodos.

El primero consiste en editar la quota de cada usuario. Por ejemplo, para editar la cuota del usuario antonio, se ejecuta desde el usuario root la instrucción:

edquota -u antonio

Quotas for user antonio:
/dev/hda3: blocks in use:15542,limits (soft=0,hard=0)
inodes in use: 2139, limits (soft = 0, hard = 0)

El sistema de cuotas de Linux permite limitar el número de bloques y el número de i-nodos que un usuario puede tener. Los valores a modificar son los límites que están puestos entre paréntesis (que ahora valen 0). Ahí se puede especificar cualquier cantidad (en Kbytes). Por ejemplo, para limitar la cuota de disco del usuario antonio a 1 Mb, se pondría lo siguiente:

Quotas for user antonio:
/dev/hda7:blocks in use:18542,limits (soft=1024,hard=1024)
inodes in use: 1139, limits (soft = 0, hard = 0)

El límite soft es un límite de aviso y el límite hard es un límite insalvable, es decir, el sistema ya no le asigna más espacio.

De una forma análoga, podríamos modificar la cuota de espacio asignada al grupo users con:

edquota -g users”[5]

1.6 Normas prácticas para aumentar la seguridad

Como administradores del sistema debemos tomar en cuenta ciertas normas que nos ayudaran a que nuestro sistema no este expuesto ataques, aunque Linux tiene valores por defecto que sirven para cubrir algunas necesidades normales.

- “ nosuid, nodev, noexec. Salvo casos excepcionales, no debería haber ninguna razón para que se permita la ejecución de programas SUID/SGID en los directorios home de los usuarios. Esto lo podemos evitar usando la opción `nosuid' en el fichero /etc/fstab para las particiones que tengan permiso de escritura por alguien que no sea el root. También puede ser útil usar `nodev' y `noexec' en las particiones de los directorios personales de los usuarios y en /var, lo que prohíbe la creación dispositivos de bloque o carácter y la ejecución de programas.
- **Sistemas de ficheros en red**

  Si exporta sistemas de archivos vía NFS, esté seguro de configurar /etc/exports con los accesos lo más restrictivos posibles. Esto significa no usar plantillas, no permitir acceso de escritura a root y montar como sólo lectura siempre que sea posible.

- **umask**

  Configure la máscara de creación de ficheros para que sea lo más restrictiva posible. Son habituales 022, 033, y la más restictiva 077, y añadirla a /etc/profile.

---

[5] http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html

La instrucción umask se puede usar para determinar el modo de creación de ficheros por defecto en su sistema. Es el complemento octal a los modos de fichero deseado. Si los ficheros se crean sin ningún miramiento de estado de permisos, el usuario, de forma inadvertida, podrá asignar permisos de lectura o escritura a alguien que no debería tenerlos. De forma típica, el estado de umask incluye 022, 027 y 077, que es lo más restrictivo. Normalmente umask se pone en /etc/profile y por tanto se aplica a todos los usuarios del sistema. Por ejemplo, puede tener una línea parecida a la siguiente:

```
# Pone el valor por defecto de umask del usuario
umask 033
```

Esté seguro de que el valor umask de root es 077, lo cual desactiva los permisos de lectura, escritura y ejecución para otros usuarios, salvo que explícitamente use chmod(1).

Si está usando , y utiliza su esquema de creación de identificador de grupos y usuarios (User Private Groups), sólo es necesario usar 002 para umask. Esto se debe a que al crear un usuario se crea un grupo exclusivo para ese usuario.

- **Limitar recursos**

Ponga el límites al sistema de ficheros en lugar de 'ilimitado' como está por defecto. Puede controlar el límite por usuario utilizando el módulo PAM de límite de recursos y /etc/pam.d/limits.conf. Por ejemplo, los límites para un grupo `users' podría parecer a esto:

```
@users hard core 0
@users hard nproc 50
@users hard rss 5000
```

Esto dice que se prohíba la creación de ficheros core, restringe el número de procesos a 50, y restringe el uso de memoria por usuario a 5M.

- **wtmp, utmp**

  Los ficheros /var/log/wtmp y /var/run/utmp contienen los registros de conexión de todos los usuarios de su sistema. Se debe mantener su integridad, ya que determinan cuándo y desde dónde entró en su sistema un usuario o un potencial intruso. Los ficheros deberían tener los permisos 644, sin afectar a la normal operación del sistema.

- **Sticky bit**

  El sticky bit se puede usar para prevenir borrados accidentales o proteger un fichero para sobreescritura. También previene que alguien cree enlaces simbólicos a un fichero, que ha sido el origen de ataques basados en el borrado de los ficheros /etc/passwd o /etc/shadow.

  SUID y SGID

  Los ficheros SUID y SGID de su sistema son potenciales riesgos de seguridad y deberían ser controlados. Como estos programas garantizan privilegios especiales al usuario que los ejecuta, es necesario estar seguro que no hay instalados programas inseguros. Un truco favorito de los crackers es explotar programas con el bit SUID, y entonces dejar un programa SUID como puerta trasera para entrar la próxima vez, incluso aunque el agujero original ya esté tapado.

  Encuentre todos los programas SUID/SGID de su sistema y mantener la pista de lo que son, para que esté prevenido de cualquier cambio que podría indicar un potencial intruso. Use la siguiente instrucción para localizar todos los programas SUID/SGID en su sistema:

  ```
  root# find / -type f \( -perm -04000 -o -perm -02000 \)
  ```

  Incluso podría crear una base de datos de progrmas SUID con

  ```
  root# find / -type f \( -perm -04000 -o -perm -02000 \)>/var/suid
  ```

y posteriormente verificar si ha aparecido alguno nuevo con el siguiente guión:

```
for fich in `find / -type f \( -perm -04000 -o -perm -02000 \)`
do
if ! grep $fich /var/suid
then
echo "$fich es un nuevo fichero con SUID"
fi
done
echo "Actualiza la base de datos si es necesario"
```

Puede eliminar los permisos SUID o SGIG de un programa con chmod(1), y siempre puede devolverlo a su estado original si piensa que es absolutamente necesario.

- Comprobar que cualquier programa con el bit SUID activo no tenga ningún desbordamiento de buffer (conocido). No asignarlo jamás si el programa permite salir a la shell.

- **Permisos de escritura**

  Los ficheros con permiso de escritura global, particularmente los del sistema, pueden ser un agujero de seguridad si un cracker obtiene acceso a su sistema y los modifica. Además, los directorios con permiso de escritura global son peligrosos, ya que permiten a un cracker añadir y borrar los ficheros que quiera. Para localizar los ficheros con permiso de escritura global, use la siguiente instrucción:

  ```
  root# find / -perm -2 -print
  ```

  y esté seguro de saber por qué tienen esos permisos de escritura. En el curso normal de una operación los ficheros tendrán permisos de escritura, incluidos algunos de /dev y enlaces simbólicos.

- **Ficheros extraños**

  Los ficheros sin propietario también pueden ser un indicio de que un intruso ha accedido a su sistema. Puede localizar los ficheros de su sistema que no tienen propietario o que no pertenecen a un grupo con la instrucción:

  root# find / -nouser -o -nogroup -print

- **Ficheros peligrosos**

  La localización de ficheros .rhosts debería ser uno de los deberes regulares de la administración de su sistema, ya que estos ficheros no se deberían permitir en sus sistema. Recuerde que un cracker sólo necesita una cuenta insegura para potencialmente obtener acceso a toda su red. Puede localizar todos los ficheros .rhosts de su sistema con la siguiente instrucción:

  root# find /home -name .rhosts -print

- **Permisos**

  Finalmente, antes de cambiar permisos en cualquier sistema de ficheros, esté seguro de que entiende lo que hace. Nunca cambie permisos de un fichero simplemente porque parezca la forma fácil de hacer que algo funcione. Siempre debe determinar porqué el fichero tiene esos permisos y propietario antes de modificarlos. “[6]

1.7 Resumen

En este capitulo se abordo la Seguridad en el Sistema de Archivos en el cual se expuso como debería realizar la estructuración jerárquica del árbol de directorios. También se presentó una herramienta muy útil para la revisión de la integridad de archivos como lo es Tripwire y por último se puso a consideración el control de

[6] http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html

espacio de almacenamiento para el usuario o grupo esto como una medida para controlar posibles problemas de seguridad.

CAPITULO II

2. SEGURIDAD FISICA

2.1 Introducción

La seguridad física es algo que debe tenerse muy en cuenta para nuestros sistemas. Debemos tomar en consideración que personas tienen acceso físico a las máquinas y si realmente deberían acceder.

El sistema operativo Linux proporciona los niveles exigibles de seguridad física como son:

- Un arranque seguro
- Posibilidad de bloquear las terminales
- Las capacidades de un sistema multiusuario real

2.2 Seguridad en el arranque

Si tratamos los aspectos de la seguridad desde la base, hemos de considerar al arranque del ordenador. Es evidente que podemos montar un servidor de manera totalmente segura, pero que si cualquiera puede acceder a la bios de esa maquina, esta se convertirá en un sistema potencialmente inseguro ya que cualquiera podrá modificar los parámetros de la maquina dejándola inutilizable o dejándola en una situación que permita acceder al sistema con total impunidad.

Es un caso clásico de sistema inseguro aquel en el cual la bios permite arrancar de disquete, de este modo cualquiera podría arrancar con un disquete del sistema operativo y montando las particiones necesarias, modificar datos como el fichero de passwords, el fichero shadow, etc...

2.2.1 LILO

“Lilo es el primer paso por el que atraviesa un sistema Linux en su arranque, en este se pueden introducir parámetros que le serán pasados al kernel del sistema operativo y que pueden influir decisivamente en el funcionamiento del sistema. Para que esto no suceda hemos de intentar que el retardo entre la carga de lilo y la del sistema operativo sea la mínima para que nadie pueda introducir parámetros innecesarios

En el caso de que sea necesario introducir parámetros en lilo, este puede obligar a introducir una password para hacerlos efectivos, a continuación presentamos algunas líneas del fichero lilo.conf .

```
delay=X
restricted
password=algun_password
```

El delay nos dará el tiempo que tarda lilo en comenzar el arranque, es aconsejable que sea poco o ninguno si no se va a producir selección en los arranques o no hay que pasar parámetros. El nivel de seguridad C2 que corresponde con los sistemas *nix habituales obliga a que no haya retardo en el arranque.

El parámetro "restricted" nos impone la condición de teclear un password cuando vaya a ser introducido algún parámetro impidiendo de esta manera que cualquiera puede cambiar el arranque de nuestro sistema.

Con "password" se exige un password para permitir el arranque del sistema.”[7]

2.3 Bloqueo de la consola

“En los entornos Unix es conocido el truco de ejecutar en una terminal, que alguien ha dejado inocentemente abierto, un guión que simule la pantalla de presentación al sistema. Entonces un usuario incauto introducirá su nombre y clave, que quedarán a merced del autor del engaño.

[7] http://david.f.v.free.fr/ponencias/Seguridad_Basica_en_Linux/seguridad-2.html#ss2.2

Si se aleja de su máquina de vez en cuando, estaría bien poder bloquear su consola para que nadie pueda manipularla o mirar su trabajo. Dos programas que hacen esto son xlock y vlock.

xlock bloquea la pantalla cuando nos encontramos en modo gráfico. Está incluido en la mayoría de las distribuciones Linux que soportan X. En general puede ejecutar xlock desde cualquier xterm de su consola y bloqueará la pantalla de forma que necesitará su clave para desbloquearla.

vlock es un simple programa que le permite cerrar alguna o todas las consolas virtuales de su máquina Linux. Puede bloquear sólo aquélla en la que está trabajando o todas. Si sólo cierra una, las otras se pueden abrir y utilizar la consola, pero no se podrá usar su vty hasta que no la desbloquee.

Cabe indicar que, bloquear una consola prevendrá que alguien manipule su trabajo, pero no previene el reinicio de la máquina u otras formas de deteriorar su trabajo."[8]

2.4 Recomendaciones

1. Uso del equipo por personal autorizado.
2. Solo podrá tener acceso al equipo aquel personal que cuente con conocimientos mínimos sobre computación.
3. Poder contar con más de un servidor de base de datos lo cual asegure la integridad total de la información.
4. Ubicación de las instalaciones que cumplan con normas internacionales de calidad (ISO 9000).
5. Control de alarma la cual notifique en todo momento sobre la integridad física del sistema.

[8] http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html

2.5 Resumen

En este capítulo se dan a conocer los niveles exigibles de Seguridad Física que tiene Linux. Adicionalmente se dan unas recomendaciones sobre como mantener seguro los equipos para asegurar la integridad física del sistema.

CAPITULO III

3. SEGURIDAD LOCAL

3.1 Introducción

Una de las primeras cosas que intentan los intrusos de un sistema como vía para explotar la cuenta de root es obtener acceso a una cuenta de usuario local; utilizando las posibles debilidades del sistema: programas con errores, configuraciones deficientes de los servicios o el descifrado de claves cifradas. Incluso se utilizan técnicas denominadas "ingeniería social", consistentes en convencer a ciertos usuarios para que suministren una información que debiera ser mantenida en secreto, como sus nombres de usuario y contraseñas. Con una seguridad local, pueden entonces "mejorar" su acceso de usuario normal a acceso de root usando diversos bugs y servicios locales pobremente configurados. Si te aseguras de que tu seguridad local es adecuada, entonces el intruso tendrá que superar otro obstáculo.

En este aspecto de la seguridad, Linux dispone de todas las características de los sistemas Unix: un control de acceso a los usuarios verificando una pareja de usuario y clave; cada fichero y directorio tienen su propietario y permisos.

3.2 Cuentas de Usuario y Grupo

"Cada usuario del sistema está definido por una línea en el fichero /etc/passwd y cada grupo por otra línea en el fichero /etc/group. Cada usuario pertenece a uno o varios grupos y cada recurso pertenece a un usuario y un grupo. Los permisos para un recurso se pueden asignar al propietario, al grupo y a otros (resto de los usuarios). Ahora bien, para mantener un sistema seguro, pero funcional, tendremos que realizar las combinaciones necesarias entre el propietario y grupo de un recurso con los permisos de los propietarios, grupos y otros. Por ejemplo, la unidad de disco flexible tiene las siguientes características:

brw-rw-r-- 1 root floppy 2,0 may 5 1998 /dev/fd0

- Propietario: root con permiso de lectura y escritura.

- Grupo: floppy con permiso de lectura y escritura.
- Otros: resto de los usuario con permiso de lectura.

Cuando queramos que un usuario pueda escribir en la unidad de disco, sólo tendremos que incluirlo en el grupo floppy. Cualquier otro usuario que no pertenezca al grupo floppy (salvo root) sólo podrá leer el disquete. “[9]

“Asegúrate de que provees cuentas de usuario sólo con los requerimientos mínimos para la tarea que necesiten hacer. Por ejemplo si proporcionas a tu hijo (10 años) una cuenta, podrías querer que él sólo tenga acceso a un procesador de textos o un programa de dibujo, pero que le sea imposible borrar datos que no son suyos.

Algunas buenas reglas empíricas sobre permitir a otra gente acceso legítimo a tu máquina Linux:

- Darles la cantidad mínima de privilegios que necesiten.
- Ser consciente de desde cuándo/dónde se conectan o se deberían estar conectando.
- Asegúrate de remover las cuentas inactivas.
- El uso del mismo ID de usuario en todos los computadores y redes es aconsejable para facilitar el mantenimiento del recuento, así como para permitir un análisis más fácil de los datos de log.
- La creación de ID de usuario de grupo debería estar absolutamente prohibida. Las cuentas de usuario también proporcionan transparencia y esto no es posible con cuentas de grupo.

Muchas cuentas de usuario local que se usan en los compromisos de seguridad son las que no han sido usadas durante meses o años. Dado que nadie está usándolas, proporcionan el vehículo ideal de ataque. “[10]

[9] http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html
[10] http://www.geocities.com/galinux2000/numero1/doc-es/seguridadelinuxcomo.htm

3.3 Seguridad de las Claves

"Los passwords en Linux como en cualquier sistema *nix, son la señal univoca que nos identifica como un usuario frente al sistema operativo, una vez hayamos introducido la password correcta el sistema nos asignará nuestro uid que nos identificará como nosotros ante el mismo.

Existe una serie de recomendaciones en cuanto a las passwords tanto de cara a los administradores como a los usuarios:

De cara a los administradores:

- Preocupación por los passwords ya que un sistema con passwords inseguros puede provocar algún incidente de seguridad grave.
- Es recomendable realizar algún test sobre las passwords del sistema al ser estas introducidas por si son fácilmente "adivinables", tests del tipo de comprobar que tiene un mínimo de caracteres, algún signo de puntuación, etc...
- Puede ser recomendable en algún caso realizar ataques "blandos" contra el fichero de passwords del sistema para comprobar que numero de passwords son inseguras.
- En algunos sistemas puede ser conveniente hacer que las claves tengan un tiempo de "expiración" y que obliguen al usuario a que sean cambiadas en un periodo de tiempo limitado, de este modo se consigue que una clave que haya sido descubierta no tenga periodo de validez ilimitada
- Existe una alternativa para asegurar que las passwords de nuestros usuarios son seguros y es generando de manera aleatoria la clave. Este método es un arma de doble filo ya que si bien conseguimos una clave segura también es una clave complicada de cara al usuario y por tanto una clave que será potencialmente apuntada o guardada en cualquier otro medio. Para que nos molestamos en encriptar nuestras claves si permanecen en algún lugar escritas.

De cara al usuario:

- Nunca poner fechas relevantes, palabras que significan algo para nosotros o cualquier término relacionado con nuestra persona: nombres, apellidos, nombres de familiares, fecha de nacimiento, etc...
- Nunca poner expresiones conocidas: "abretesesamo", "abracadabra".
- Es conveniente que la clave vaya acompañado de signos de puntuación, números, etc. Por ejm: "yo2./er" , "io2,,er."
- Derivado directamente del modo de ataques a passwords, es conveniente no poner como clave ninguna palabra que tenga significado o se encuentre en el diccionario de ninguna lengua (evitando así una gran parte del peligro que entrañan los ataques por diccionario).
- NUNCA apuntar nuestra clave en un papel, o en cualquier lugar ya que si bien puede ser muy segura cualquier persona que tenga acceso a este lugar donde está escrita en claro, no tendrá ni que molestarse en desencriptarla.
- Es muy aconsejable utilizar reglas nemotécnicas de las cuales salgan passwords que únicamente tienen sentido para nosotros de manera que no se encuentren en ningún diccionario pero sin embargo sean fáciles de recordar para nosotros."[11]

Para comprobar que estos requisitos se cumpla en nuestro sistema, podemos usar los mismos mecanismos que utilizan los atacantes. Existen varios programas que van probando varias palabras de diccionario, claves habituales y combinaciones de caracteres, que son cifrados con el mismo algoritmo que usa el sistema para mantener sus claves; después son comparadas con el valor de la clave cifrada que queremos averiguar hasta que el valor obtenido de un cifrado coincide con una clave cifrada. Posteriormente notificaremos al usuario que su clave es débil y le solicitaremos que la modifique.

"Los ficheros de passwords en Linux residen habitualmente en el directorio /etc/passwd (aunque existen excepciones como veremos con shadow). Es de hacer notar que el cifrado de las claves en *nix es "de un solo sentido" es decir, teniendo la clave y el algoritmo de cifrado podremos generar la clave cifrada, pero teniendo el algoritmo y la clave cifrada no

[11] http://david.f.v.free.fr/ponencias/Seguridad_Basica_en_Linux/seguridad-3.html

podremos generar la clave sin cifrar. Es interesante tener esto en cuenta ya que los ataques al fichero de claves se realizan teniendo presente este modo de cifrado.

Esta sería una línea típica de un /etc/passwd (sin shadow) :

david:aaBaCDeg:1000:1000:David Fernandez Vaamonde,,,:/home/david:/bin/bash

El proceso de validación de una password es el siguiente:

- Se toma el login y la password (sin echo al terminal).
- La palabra clave se encripta con el algoritmo de encriptación utilizado para las passwords
- Se compara esa password encriptada con la existente
- Si coinciden se valida, si no se descarta."

"Cualquier usuario del sistema tiene permiso de lectura sobre este fichero, pudiendo de esta manera llegar hasta un hacker que cree páginas web que exploten estos agujeros.

Entonces puede parecer a primera vista que nos encontramos con un grave agujero de seguridad. El atacante, una vez obtenido el fichero `/etc/passwd` no tiene más que ejecutar su programa revientaclaves favorito y sentarse a esperar hasta que empiecen a aparecer nombres de usuario con sus respectivas contraseñas, algo que suele pasar muy rápidamente. Con suerte, si el administrador es ingenuo o dejado, incluso dará con la clave del root, abriéndosele así las puertas a la máquina objetivo. Para solucionar esta vulnerabilidad, podemos recurrir a contraseñas en la sombra (shadow passwords), un mecanismo consistente en extraer las claves cifradas del fichero /etc/passwd y situarlas en otro fichero llamado /etc/shadow, que sólo puede leer el root y dejar el resto de la información en el original /etc/passwd. Este fichero es de lectura y escritura para el root y de escritura únicamente para el grupo shadow. En este fichero se almacenan las passwords encriptadas de manera que únicamente el root pueda acceder a ellas, de este modo se evita mostrar a cualquier persona las passwords en claro para que pueda realizar un ataque contra el fichero de claves del sistema a su antojo. Este es el aspecto que presenta una línea del fichero /etc/passwd en un sistema con passwords en shadow:

david:x:1000:1000:David Fernandez Vaamonde,,,:/home/david:/bin/bash

Y aquí esta la lineal del /etc/shadow correspondiente:

david:njAd87ds.adferw:10876:0:99999:7:::

El formato del fichero /etc/shadow es similar al siguiente:

usuario : clave : ultimo : puede : debe : aviso : expira : desactiva : reservado

- usuario: El nombre del usuario.
- clave: La clave cifrada
- ultimo: Días transcurridos del último cambio de clave desde el día 1/1/70
- puede: Días transcurridos antes de que la clave se pueda modificar.
- tiene: Días transcurridos antes de que la clave tenga que ser modificada.
- aviso: Días de aviso al usuario antes de que expire la clave.
- expira: Días que se desactiva la cuenta tras expirar la clave.
- desactiva: Días de duración de la cuenta desde el 1/1/70.
- reservado: sin comentarios.

Un ejemplo podría ser:

Maria : gEvm4sbKnGRlg : 10627 : 0 : 99999 : 7 : -1 : -1 : 134529868”[12]

3.4 El bit SUID/SGID

“En muchas ocasiones un proceso necesita ejecutarse con unos privilegios mayores (o menores) que el usuario que lo lanzó. Por ejemplo, un usuario puede modificar su propia clave con la instrucción **passwd**, pero esto implica modificar el fichero **/etc/passwd**, y para esto un usuario "de a pie" no tiene permiso. Esto se puede solucionar activando el bit SUID de la instrucción **passwd** (nótese que cuando esto sucede, la x de ejecutable pasa a ser una s):

[12] http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html

ls -la /usr/bin/passw*

-r-sr-xr-x 1 root bin 15613 abr 27 1998 /usr/bin/passwd

Esto quiere decir que cuando se ejecute, el proceso correspondiente va a tener los privilegios del propietario de la instrucción (es decir, el root), no del usuario que lo lanzó. En otras palabras, el proceso generado por passwd pertenece a root. A primera vista, esto puede parecer una seria brecha de seguridad. Y lo es. Si el programa funciona correctamente, no tiene por qué dar problemas; pero pequeños defectos en el programa pueden ser utilizados por alguien para tratar de ejecutar otro código distinto con los privilegios de este proceso. El método suele ser el desbordamiento de la pila (buffer overflow).

Cualquier atacante que haya entrado en un sistema de forma ilegítima intentará dejar una shell con el bit SUID para mantener ese nivel de privilegio cuando vuelva a entrar en el sistema.

SGID es lo mismo que SUID, pero aplicado al grupo.

Así pues, tenga cuidado con los programas con el bit SUID/SGIG. Puede encontrarlos con

```
root# find / -type f \( -perm -04000 -o -perm -02000 \) -print
```

Tenga en cuenta que algunos programas (como passwd) tienen que tener el bit SUID. Compruebe en los lugares habituales, que ninguno de los programas propiedad del root o SUID que utiliza en su sistema, tiene un fallo de seguridad conocido que pueda ser explotado.

Nunca debe permitir que quede una shell SUID corriendo en el sistema. Si el root deja desatendida la consola durante unos instantes (recuerde, debe utilizar siempre xlock).”[13]

[13] http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html

3.5 Resumen

La Seguridad Local tiene que ver con los usuarios y grupos en este capitulo se especifica como se debería asignar o crear las contraseñas para evitar que estas claves débiles puedan ser descifradas por atacantes y explotar la cuenta de root.

También hay que tener cuidado con el bit SUID/SGID ya que cualquier atacante que haya ingresado al sistema dejara un shell con el bit SUID para mantener el mismo privilegio y volver ingresar

CAPITULO IV

4. SEGURIDAD DEL ROOT

4.1 Introducción

El mayor enemigo del sistema es el propio administrador, sí, tiene todos los privilegios y cualquier acción puede ser irreversible y hacerle perder posteriormente mucho más tiempo que el que hubiera perdido por realizar las tareas de forma segura. Puede borrar cualquier fichero e incluso destruir el propio sistema, mientras que un usuario "normal" sólo puede perjudicarse a sí mismo. Por estos motivos, conseguir privilegios de root es la meta de cualquier ataque.

A continuación presentaremos unas normas que le serán de gran utilidad al administrador:

- "No use la cuenta de root por norma. Evítela. Intente primero cualquier acción como un usuario normal, y si ve que no tiene permiso, piense porqué y use la instrucción su si es necesario.
- Ejecute las instrucciones de forma segura verificando previamente la acción que va a realizar. Por ejemplo si quiere ejecutar rm borrar.*, ejecute primero ls borrar.* y si es lo que pretende modifique el mandato y ejecútelo.
- Ciertos mandatos admiten una opción (-i) para actuar de forma interactiva. Actívela, si no lo está ya añadiendo estas líneas a su fichero de recursos par la shell:
    - alias rm='rm -i'
    - alias cp='cp -i'
    - alias mv='mv -i'

    Siempre puede evitar estas preguntas, a veces incordiosas, con el mandato yes, cuando esté seguro de la operación que está realizando:

    yes s|rm borrar.*

- Como ya deben saber, el directorio actual no está, por defecto, en la ruta de búsqueda de ejecutables (PATH). Esto garantiza que no lanzaremos, sin darnos cuenta, un ejecutable que esté en el directorio actual llamado, por ejemplo ls.
- Evite que la clave del root viaje por una red sin cifrar. Utilice ssh u otro canal seguro.
- Limite los terminales desde los que se puede conectar root. Es preferible limitarlo a la consola del sistema. Esto se puede decidir en /etc/securetty. Si necesita una sesión remota como root, entre como usuario normal y luego use su.
- Actúe de forma lenta y meditada cuando sea root. Sus acciones podrían afectar a un montón de cosas.

4.2 Determinar los Servicios Activos

Debe desactivar todos los servicios que no vaya a prestar, en particular revise los ficheros /etc/inittab, /etc/inetd.conf y los demonios que se lanzan durante el arranque. Si no está realmente seguro de lo que hace, mejor no haga nada; las distribuciones más modernas incorporan unos mínimos de seguridad aceptables para un usuario medio.

No tiene sentido tener abierto un servicio del que no va a hacer uso ningún usuario legal. Puede que esté consumiendo recursos de su sistema para ofrecer a algún atacante la posibilidad de violarlo.

Puede usar un analizador de puertos para ver qué parte de su sistema es visible desde el exterior. Existen utilidades como SATAN**,** Nessus o nmap que realizan esta labor. “[14]

4.3 Mantener el Software Actualizado

El software que acompaña a Linux por lo menos la gran mayoría es de código fuente público, como el propio núcleo; siendo esto una garantía de seguridad, ya que cientos de expertos analizan el código para detectar alguna falla que poder publicar en las listas de

[14] http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html

correo sobre seguridad más prestigiosas, y se corrigen con gran rapidez. De esta manera se garantiza un software de calidad y no seguridad aparente. Por otro lado nos obliga a ir sustituyendo las versiones defectuosas por las corregidas y que mejoran las prestaciones. En cualquier sistema operativo, mantener un software que ha demostrado tener fallos supone asumir un riesgo innecesario.

4.4 Copias de Seguridad

“Existen acontecimientos de los que nos puede resultar muy difícil protegernos como son los desastres naturales, únicamente podremos seguir una serie de pasos para evitar que su incidencia sea lo menor posible. La mejor solución es mantener un buen conjunto de copias de seguridad sobre toda la información necesaria del sistema. Hay que pensar que las copias de seguridad no sólo nos protegen de desastres naturales, también de los desastres que pueda ocasionar algún intruso en nuestro sistema, de cualquier ataque a la disponibilidad o integridad de la información del sistema.

Si los datos tienen un tamaño inferior a 650Mb, puede ser una buena opción grabarlos en un CD, bien permanente (ya que es más difícil de falsificar con posterioridad, y si están almacenados de forma adecuada pueden durar mucho tiempo) o regrabable. Las cintas y otros medios sobre los que se puede escribir deberían protegerse tan pronto como se completa la copia y se verifica para evitar la falsificación. Tenga cuidado y almacene su copia de seguridad en un sitio seguro. Una buena copia de seguridad le asegura que tiene un buen punto desde el que restaurar su sistema.

Hay que insistir en la seguridad de las copias de seguridad. Si las copias de seguridad no están almacenadas en un sitio seguro, puede que el posible intruso no tenga necesidad de idear métodos sofisticados para obtenerla, si le basta con copiar o sustraer un CD.

4.4.1 Características de las copias de seguridad

Cuando se realice una copia de seguridad es conveniente seleccionar un método que garantice la conservación de las características de la información como son derechos y permisos. Si realizamos una copia de seguridad de una forma o sobre un soporte que no

contemple esta posibilidad, si tenemos que restaurar los datos sobre el sistema el resultado sobre la seguridad y funcionalidad globales puede ser impredecible.

4.4.2 Secuencia de Copias

Es necesario tener un política de copias de seguridad adecuada a las características de la entidad que estamos gestionando. Quizás el mejor método es el de rotación de cintas. Pasamos a verlo con un ejemplo.

Un ciclo de seis cintas es fácil de mantener. Esto incluye cuatro cintas para la semana, una cinta para cada viernes y una cinta para los viernes impares. Se realiza una copia incremental cada día, y una copia completa en la cinta adecuada de cada viernes. Si hace algún cambio importante o añade datos importantes a su sistema también sería adecuado efectuar una copia.

4.4.3 Copiar las Bases de Datos del Sistema

Existe cierta información del sistema que es imprescindible para su correcto funcionamiento. Es conveniente tener una copia de estos ficheros en una ubicación segura. En particular resulta conveniente tener una copia del contenido del directorio /etc. También hay que mantenerla en lugar seguro, ya que tiene copias de los ficheros /etc/passwd y /etc/shadows, entre otros que puedan contener claves de usuarios que no están cifradas.

También en cada sistema se puede tener una base de datos de las aplicaciones que hay instaladas en el servidor. Cada distribución dispone de alguna herramienta que nos realiza el mantenimiento de la base de datos a la misma vez que instala o desinstala aplicaciones. La pérdida de esta base de datos nos haría perder el control sobre qué aplicaciones tenemos instaladas.

En muchas situaciones también será necesario tener copia de seguridad de los ficheros de registro de incidencias, para tener constancia de las distintas actividades del sistema."[15]

---

[15] http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html

4.5 Búsqueda de vulnerabilidades en el sistema Operativo usando y configurando Nessus

"Nessus es uno de los escaners de seguridad más utilizados en la actualidad. Un scanner de seguridad intenta descubrir de forma automática que posibles errores de seguridad pueden existir en un sistema. Para ello va ejecutando una serie de ataques al objetivo, comprobando la respuesta de este, sabrá si este es vulnerable o no. Como resultado obtendremos una lista de los problemas que puede haber en ese sistema. Como siempre esto puede utilizarse de forma muy diferente: para parchear el sistema o para efectuar un ataque contra él.

Nessus está basado en una arquitectura de plugins que lo hacen muy potente y fácilmente actualizable. Así cada plugin contiene la secuencia de acciones del ataque y como responderá un sistema vulnerable. Cuando aparece un nuevo ataque solo hemos de instalar el nuevo plugin y así poder comprobar nuestro sistema utilizando siempre la misma herramienta."[16]

4.5.1 Instalación

Nessus es un escáner de seguridad libre disponible en http://www.nessus.org/; luego tenemos que acceder a la opción DOWNLOAD, en la parte de seleccionar el producto a descargar seleccionamos el Nessus 2.2.6 installer (all Unix systems); desplegándose una ventana que es la aceptación de la licencia y damos clic en aceptar pidiéndonos la dirección de email para enviarnos el código de activación, y se procede a la descarga.

Para la instalación ejecutamos las siguientes instrucciones:

mount –o remount, exec /tmp

./nessus-installer-2.2.6.sh

Congratulations ! Nessus is now installed on this host

. Create a nessusd certificate using /usr/local/sbin/nessus-mkcert

[16] http://linux.ideando.net/articulos/hs/hs11.htm

. Add a nessusd user use /usr/local/sbin/nessus-adduser

. Start the Nessus daemon (nessusd) use /usr/local/sbin/nessusd -D

. Start the Nessus client (nessus) use /usr/local/bin/nessus

. To uninstall Nessus, use /usr/local/sbin/uninstall-nessus. Remember to invoke '/usr/local/sbin/nessus-update-plugins'

periodically to update your list of plugins

A step by step demo of Nessus is available at :

http://www.nessus.org/demo/

Press ENTER to quit

You have new mail in /var/spool/mail/root

4.5.2 Configuración

Para la configuración tenemos que realizar los siguientes pasos:

1) [root@localhost /]# /usr/local/sbin/nessus-mkcert

Creation of the Nessus SSL Certificate

This script will now ask you the relevant information to create the SSL
certificate of Nessus. Note that this information will *NOT* be sent to
anybody (everything stays local), but anyone with the ability to connect to your
Nessus daemon will be able to retrieve this information.
CA certificate life time in days [1460]: 60
Server certificate life time in days [365]: 60
Your country (two letter code) [FR]: EC
Your state or province name [none]: AZUAY
Your location (e.g. town) [Paris]: Cuenca
Your organization [Nessus Users United]: Home

```
--------------------------------------------------------------------------------
Creation of the Nessus SSL Certificate
--------------------------------------------------------------------------------
Congratulations. Your server certificate was properly created.
/usr/local/etc/nessus/nessusd.conf updated
The following files were created :
. Certification authority :
  Certificate = /usr/local/com/nessus/CA/cacert.pem
   Private key = /usr/local/var/nessus/CA/cakey.pem
. Nessus Server :
 Certificate = /usr/local/com/nessus/CA/servercert.pem
 Private key = /usr/local/var/nessus/CA/serverkey.pem
 Press [ENTER] to exit
2) [root@localhost /]# /usr/local/sbin/nessus-adduser
Using /var/tmp as a temporary file holder
Add a new nessusd user
----------------------
Login : olga
Authentication (pass/cert) [pass] : pass
Login password :
Login password (again) :
User rules
nessusd has a rules system which allows you to restrict the hosts
that olga has the right to test. For instance, you may want
him to be able to scan his own host only.
Please see the nessus-adduser(8) man page for the rules syntax
Enter the rules for this user, and hit ctrl-D once you are done :
(the user can have an empty rules set)
Login            : olga
Password         : ***********
DN               :
```

Rules :

Is that ok ? (y/n) [y] y

user added.

3) [root@localhost /]# /usr/local/sbin/nessusd -D

All plugins loaded

4.6 Normas para Aumentar la Seguridad

- "Suscribirse a las listas de correo de alertas de seguridad para estar actualizado.
- Prestar atención a los ficheros de registro.
- Actualizar el software inseguro.
- Verificar regularmente la integridad de los ficheros con algún software como tripwire.
- Tener unas copias de seguridad adecuadas.
- Utilizar PGP o GnuPG para garantizar la autenticidad y la privacidad.
- Verificar con periodicidad los puertos de los equipos.
- Revisar periódicamente las cuentas de usuario.
- Asignar cuotas de uso de recursos del sistema.
- Mantener los terminales seguros.
- Asegurarse de tener claves sólidas.
- Mantener el sistema de ficheros con propietarios y permisos adecuados.
- Instalar cortafuegos. "[17]

4.7 Resumen

En este Capitulo de Seguridad del Root se revisaron algunos temas que nos ayudaran a mantener la seguridad como es la Verificación de servicios activos estos deben estar claramente definidos por al administrador debido a si se mantiene servicios activos que sean innecesarios puede ser una puerta abierta para ataques. Mantener un software actualizado nos da garantía de estar protegidos debido a que existen mejoras en cada

[17] http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html

versión. También se da consejos para realizar un plan de contingencia tomando en cuenta que los respaldos por seguridad deben estar en un lugar fuera de la empresa.

CAPITULO V

5. HERRAMIENTAS DE SEGURIDAD

5.1 Introducción.

A continuación presentaremos algunas herramientas de un Sistema Linux Centos 4.2 que nos ayudaran a mantener protegido nuestro sistema de posibles intrusos, estas son:

- Nmap
- Snort
- Iptraf
- Etheral
- Iptables
- Cifrado del Disco

5.2 Escaneo de puertos:Nmap

El escaneo de puertos es una técnica usada para auditar máquinas en las que necesitamos saber que servicios se están sirviendo y de que modo. Con nmap, se conseguirá no solo saber que servicios están disponibles en las máquinas de nuestra red si no también orientación sobre su el sistema operativo que corre en ellas.

Nmap es una herramienta para el escaneo de puertos probablemente de las mejores que existen siendo software libre. Utilizando varias técnicas para realizar el escaneo de estos puertos (que son posibles canales de comunicación en potencia) tanto para TCP como para UDP.

Algunas de estas técnicas:

- TCP connect() : Usa la llamada al sistema connect para intentar una conexión a un puerto, si no es posible, ese puerto no esta abierto.

  Ej: nmap -sT localhost

- TCP SYN : Manda un paquete SYN y espera la contestación, nunca abre una conexión TCP del todo, en función de la respuesta (un ACK o un RST).

  Ej: nmap -sS localhost

- TCP FIN : Se usa para evitar que los paquetes SYN sean cortados por el firewall, se usa FIN y se espera el RST de resultado.

  Ej: nmap -sF localhost

  (únicamente funciona con sistemas UNIX)

- UDP scan : Escanea los servicios que permiten tráfico UDP.

  Ej: nmap -sU localhost

Para realizar estos escaneos, nmap comprueba todos los puertos de una maquina con estas técnicas. Esto puede llevar un tiempo considerable, de modo que nmap contempla la posibilidad de escanear únicamente puertos determinados y para eso da la oportunidad de escanear los puertos en el fichero: /etc/services

5.2.1 Instalación

Nos descargamos nmap del sitio www.insecure.org , la versión nmap-4.01-1.i386.rpm y procedimos a instalarlo con la siguiente instrucción:

rpm –Uvh nmap-4.01-1.i386.rpm

5.2.2 Ejecución de la Instrucción

[root@localhost]# nmap 192.168.55.2

Starting nmap V. 4.01 (www.insecure.org/nmap/)

Interesting ports on 192.168.55.2
(The 1667 ports scanned but not shown below are in state: closed)

Port State Service

7/tcp open echo

22/tcp open ssh
111/tcp open rpcbind
1004/tcp open unknown
12401/tcp open nessus

Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds

Una vez determinados los puertos que tenemos abiertos en la máquina, es nuestra sugerencia que procedamos a determinar cuáles de estos puertos son los que realmente usamos. En el caso de esta máquina tenemos dudas de quién está escuchando en el puerto 111 así que procedemos a determinarlo desde la máquina Linux con una instrucción llamada: **netstat**:

netstat -anp|grep 111

y nos mostrará:

[root@localhost]# netstat -anp|grep 111

tcp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN 2074/portmap
udp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN 2074/portmap

“Existen otras opciones avanzadas del nmap como:

- **P0**

  A veces sabemos que una máquina existe, pero que no está respondiendo a los pings; a veces los desarrolladores o nosotros mismos consideramos que si no respondemos a los ping nuestra máquina estará más segura. La primera labor que hace el nmap al comenzar a escanear una máquina o red es hacerle ping a los dispositivos de red, y los que responden al ping son escaneados, los otros no.

Si conocemos que un dispositivo existe pero no responde al ping, podemos usar el switch: -P0 para indicarle que sabemos que esa máquina existe pero no queremos que le haga ping, sino que escanee directamente.

Por supuesto, si la máquina no existe en verdad perderemos tiempo pues el nmap comenzará a escanear ciegamente y a esperar respuestas que no le llegarán pues la máquina no existe.

```
nmap -P0 192.168.55.2
```

Hará lo mismo que la instrucción anterior pero no se fijará si el dispositivo responde al ping o no.

- Revisando redes completas:

Esta instrucción nos permitirá revisar redes completas, tengan en cuenta que una red puede ser bien grande y hacer que demore mucho el escaneo con nmap, además si lo combinamos con -P0, podemos estar seguro que demorará unos cuantos minutos u horas:

```
nmap 192.168.55.2/24
```

Esta opción nos indica que revise la red 192.168.1.0 (el ultimo numero es ignorado) con una máscara /24 (es decir, que tome como fijo los 24 primeros bits, 3 primeros octetos, por eso el ultimo numero no tiene utilidad en este caso).

El reporte nos lo irá dando de máquina en máquina hasta que acabe de revisar la red completamente.

- Una opción muy importante el nmap es la posibilidad que tiene de determinar, dada la información del stack TCP IP, el sistema operativo que corre en la máquina que está siendo revisada. Es muy simple:

nmap -O 192.168.55.2

no sólo determinará los puertos abiertos de esa máquina, sino que también dirá el SO de ella.

A propósito: Las instrucciones pueden ser usados combinados; es decir, podemos hacer un nmap de toda la red, combinado con el switch -O y el -P0:

nmap -O -P0 192.168.55.2/24

- Determinando hosts activos en la red:
  Para esto se usa la instrucción:

  nmap -sP 192.168.55.2/24

  El cual nos indicará qué máquinas de la red han respondido al ping. Solamente eso.

- Escaneo transparente:

  Esta es una forma que se puede usar para intentar evadir sistemas de detección de intrusos en nuestra red, lo que hace es no cerrar la conexión inicial (3 way handshake) aunque ya la mayoría de los sistemas de detección de intrusos detecta este tipo de escaneo, al menos nos servirá para ver en los logs de nuestros IDS si se reporta este tipo de escaneo:

  nmap -sS 192.168.55.2

  Realizará un escaneo transparente. Si el firewall que tenemos es moderno, éste indicará que le hemos escaneado, si no reporta nada, es que no está realizando detección contra escaneos transparentes.

  nmap -sS 192.168.55.1 -p 80,110

  Esta instrucción anterior nos indicará que revise solamente los puertos 80 y 110 de forma transparente.

  nmap -sS 192.168.55.2 -D 1.1.1.1,5.5.5.5,6.6.6.6,7.7.7.7 192.168.55.2

La instrucción anterior le indicará al NMAP que falsifique la dirección de origen haciendo pasar que viene desde (1.1.1.1,5.5.5.5,6.6.6.6,7.7.7.7), algunos IDS no son capaces de lidiar con múltiples orígenes y solo reportan los 4 primeros. Si nuestro IDS solo reporta estas IP que hemos falsificado y no reporta la nuestra, es que no está teniendo una eficiente determinación de la IP agresora."[18]

5.3 Sistema de Detección de Intrusos: Snort

Snort es un detector de intrusiones de red de código abierto capaz de realizar análisis del tráfico de red en tiempo real e indicarnos posibles anomalías. Está basado en un sistema de reglas muy flexible que nos permite estar actualizados en todo momento.
Snort puede funcionar en 3 modos: sniffer, log de paquetes y IDS.

5.3.1 Instalación

Como siempre descargamos el paquete rpm de la dirección www.snort.org .Hemos de tener en cuenta que libpcap debe estar instalada.
La instalaciòn se la realiza de la siguiente manera:

```
[root@localhost ~]# cd Desktop

[root@localhost Desktop]# rpm -Uvh snort-2.3.3-1.2.el4.rf.i386.rpm

warning: snort-2.3.3-1.2.el4.rf.i386.rpm: V3 DSA signature: NOKEY, key ID 6b8d79e6

Preparing...                ########################################### [100%]

   1:snort                  ########################################### [100%]

[root@localhost Desktop]#
```

5.3.2 Ejecución de la Instrucción

- **"Modo sniffer**

  En este modo, el snort no actúa como un IDS, sino que sencillamente puede oler lo que circula por nuestra red:

[18] http://www.ernestoperez.com/documentacion/introduccion_al_nmap

`snort -v` : Imprimirá los encabezados de los paquetes TCP/IP

`snort -vd` : Imprimirá los encabezados así como los datos del paquete

`snort -vde` : Además imprimirá los encabezados de la capa Data Link

Es de hacer notar que los switches (opciones) se pueden enviar separadamente además, esta última instrucción, podía haberse invocado así:

```
snort -v -d -e
```

- **Packet logger**

  Hasta el momento hemos visto que el snort puede mostrarnos estas instrucciones a pantalla. Pero que tal si quisiéramos guardar paquetes a disco.

  ```
  snort -dev -l ./log
  ```

  Lo que hemos hecho es agregar el switch -l indicándole que guarde los datos en el directorio ./log, si este directorio no existe hay que crearlo primeramente.

  Esto creará un directorio por cada IP que escuche en la red, si deseamos guardar nuestra red interna en una sola IP, podemos agregar:

  ```
  snort -dev -l ./log -h 192.168.55.0/24
  ```

  Esto crearía un solo directorio de históricos dentro de ./log para nuestra red interna (suponiendo que es 192.168.55.0/24)

- **Modo NIDS (Network Intrusion Detection System)**

  En este modo, el snort no guarda cada paquete que circula por la red, sino sencillamente los que ameritan por su patrón.

  Ejecutemos el snort en modo IDS:
  snort -d -h 192.168.55.0/24 -l /var/log/snort -c /etc/snort/snort.conf

Procederá a guardar los eventos inusuales dentro de /var/log/snort y leerá la configuración de estos eventos en: /etc/snort/snort.conf

Si tenemos una cantidad enorme de paquetes circulando por la red, podemos indicarle al snort que ejecute en modo fast, para guardar información mínima de los paquetes y no tener cuellos de botella: snort -d -h 192.168.55.0/24 -l /var/log/snort -A fast -c /etc/snort.conf

Si queremos correr snort en modo de **demonio**, agregamos el switch -D:

snort -d -h 192.168.1.0/24 -l /var/log/snort -A fast -c /etc/snort.conf -D

En este modo, el snort comienza a correr y devuelve acceso al prompt.

5.3.3 Archivo de Configuración Snort.conf

Este archivo es el encargado de ejecutar las reglas del snort que serán usadas para detectar cualquier intento de intrusión. Las reglas las guardaremos en /etc/snort, editemos /etc/snort/snort.conf y busquemos la palabra:

RULE_PATH ../rules

Cambiémosla por: RULE_PATH /etc/snort

Reiniciemos el snort: killall -HUP snort

o mejor, matémoslo y volvámoslo a llamar:

service snort restart

Eso es todo, el snort guardara los eventos definidos en snort.conf dentro de /var/log/snort/."[19]

[19] http://www.ernestoperez.com/documentacion/instalacion_de_un_nids_snort.html

5.4 Monitoreo

5.4.1 Iptraf

Esta es una herramienta que permite el análisis de tráfico y redes es decir tener datos acerca de una importante cantidad de variables que intervienen en una conexión,(conexiones TCP, conteo de paquetes y bytes, estadísticas de las interfaces e indicadores de actividad, trafico TCP , UDP e ICMP, monitoreo de la LAN a través de las direcciones MAC, conteo de paquetes y bytes de la LAN, estadísticas de tráfico por puerto, protocolo y servicio, y muchas otras funcionalidades.

Esta herramienta esta programada usando las librerías Ncurses.

Se la recomienda especialmente para su uso remoto, ya que al contar con una interfaz de texto es muy liviano en las comunicaciones de bajo ancho de banda. Una funcionalidad interesante es la de poder crear y aplicar filtros a cada uno de los análisis y estadísticas que realiza, como así también la posibilidad de loguear en archivos la actividad registrada.

Reconoce los siguientes protocolos:

- IP
- TCP
- UDP
- ICMP
- IGMP
- IGP
- IGRP
- OSPF
- ARP
- RARP

Los paquetes que no correspondan a la lista anterior simplemente serán marcados como "non-ip".

Soporta además los siguientes tipos de interfaces.

- Local loopback
- Ethernet
- FDDI
- SLIP
- Asynchronous PPP
- Synchronous PPP sobre ISDN
- ISDN con encapsulación Raw IP
- ISDN con encapsulación Cisco HDLC

- Parallel Line IP

5.4.1.1 Instalación de Iptraf

Para instalar el Iptraf primero se tiene que descargar del sitio http://linux.maruhn.com/sec/Iptraf.html y bajarse el archivo iptraf-2.7.0-11.i386.rpm.
Luego se ejecuta la instrucción para la instalación:
rpm –Uvh iptraf-2.7.0-11-i386.rpm

5.4.1.2 Configuración de Iptraf

Para ver las estadísticas detalladas para la interface con la que te conectas a internet (Detailed interface statistics); en mi caso la interface es eth0, mira los datos que te ofrece de cómo va evoluacionando tu conexión (paquetes,, bytes, totales, de entrada, de salida, por IP, TCP, UDP, ICMP, Other IP y Non-IP), y en la parte inferior te informa sobre la media cada 5 segundos del total, la entrada y la salida (en kbs o paquetes/s).

La otra opción que se debe configurar es 'Configure' del menú principal se entra en el menú de configuración del programa. La opción 'logging' puede estar 'on' u 'off'. Si está activada significa que se está añadiendo cada cierto intervalo de tiempo los datos mostrados en la pantalla a un fichero de inventario de sucesos (usualmente lo llamamos log), para activar esta opción de 'logging' en este menú. Después pulse sobre la opción 'timers' y se te abrirá un submenú en el que tendrás que seleccionar la opción 'logging interval'. Una vez dentro, escribe 1, y acepta con enter. Lo que acabamos de hacer significa que indicamos al programa que ha de escribir en un fichero log cada minuto los datos de la pantalla, con el tipo de estadísticas seleccionado para la interfase seleccionada. El programa iptraf aún no está escribiendo ningún fichero log, falta indicarle el nombre del fichero. Vuelve al menú inicial del programa saliendo de los menús (exit) y entra de nuevo en las 'Detailed interface statistics', elige tu interfase y ahora sí que te pregunta el nombre del fichero para almacenar el log. Deje el nombre y trayectoria por defecto: /var/log/iptraf/iface_stats_detailed-eth0.log

5.4.1.3 Ejecución de Iptraf

Se pulsa en la línea de instrucciones:
[root@localhost ~]# iptraf
para su ejecución en modo consola ya que es una forma más comprensible.

Ahora lo ejecutaremos en modo background como demonio:
[root@localhost ~]# iptraf -d eth0 -B
que significa:

-d eth0: es la interfaz sobre la que escribe los datos en el log.

-B: es el tipo de estadísticas sobre las que va a llevar el log, concretamente del tipo 'detailed interface statistics'. (Nota: no uses otro tipo, porque los script que te comento después en este documento sólo funcionan con ese tipo de estadísticas).

No le pongo más parámetros, porque supongo que el intervalo de 'longeo' por defecto es 1 minuto (no existe parámetro para especificarlo), y no le especifico 'timeout' para que el demonio nunca termine su ejecución, por lo que se ejecutará de forma indefinida.

5.4.2 Ethereal

Es un analizador de protocolos con interfaz gráfica capaz de reconocer muchos protocolos distintos. Permite tanto revisar los paquetes de datos en una red activa como desde un archivo de captura previamente generado; es capaz de comprender diversos formatos de archivo propios de otros programas de captura, en particular el clásico tcpdump.

5.4.2.1 Instalación

En el menú aplicaciones, en la opción
System Settings, luego
Add/Remove packages, se abre una ventana donde buscamos y marcamos la opción System Tools y escogemos la opción Details; dentro de esto activamos las opciones:

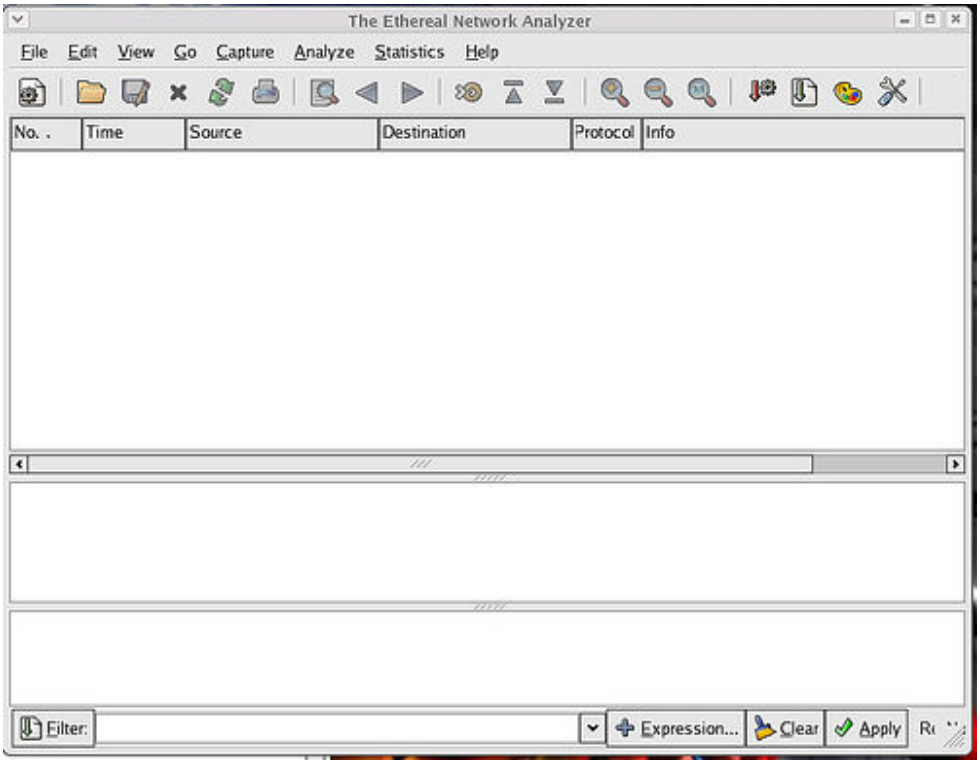
- Ethereal – Network Traffic Analyzer

- Ethereal-gnome – Red Hat Gnome Integrations for ethereal and ethereal- usermode

y pulso Close y luego Update; y saldrá una ventana especificándole que CD. debe insertar para proceder a la instalación.

5.4.2.2 Ejecución de la Instrucción

Para arrancar el programa escribimos 'Ethereal' en la línea de comandos.

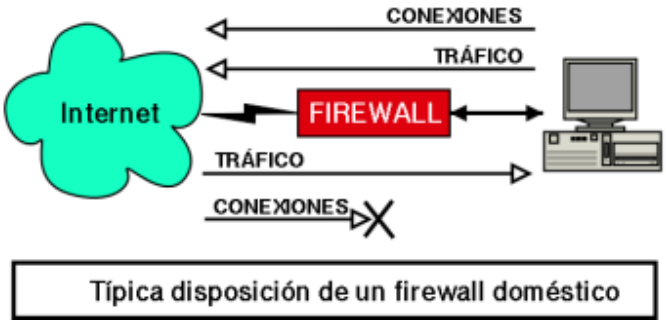


5.5 Desarrollo de un firewall usando Iptables

“Un firewall es un dispositivo que filtra el tráfico entre redes, como mínimo dos. El firewall puede ser un dispositivo físico o un software sobre un sistema operativo. En general debemos verlo como una caja con DOS o mas interfaces de red en la que se establecen una reglas de filtrado con las que se decide si una conexión determinada puede establecerse o no. Adicionalmente se pueden realizar modificaciones sobre las comunicaciones como el NAT.

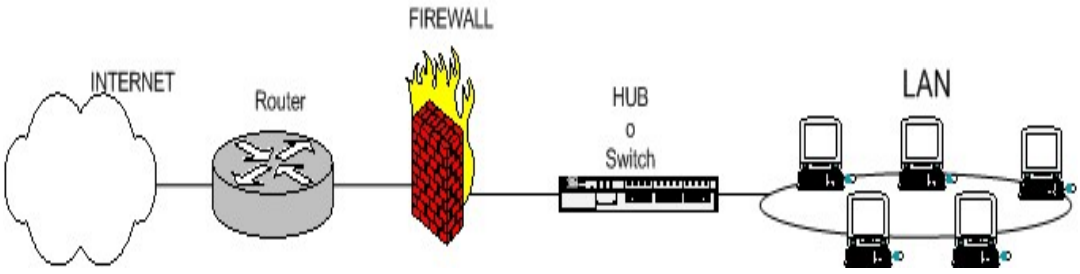
Esa sería la definición genérica, hoy en día un firewall es un hardware específico con un sistema operativo o una IOS que filtra el tráfico TCP/UDP/ICMP/../IP y decide si un paquete pasa, se modifica, se convierte o se descarta."[20]

"Esta es la configuración de un típico firewall doméstico, que permita conectarse a Internet de una manera segura y cerrar los puertos TCP y UDP que nos puedan causar problemas.



La figura 1 es un esquema típico de un firewall doméstico."[21]

"Para que un firewall entre redes funcione como tal debe tener al menos dos tarjetas de red. Esta sería la tipología clásica de un firewall:



La figura 2 es un esquema típico de firewall para proteger una red local conectada a internet a través de un router. El firewall debe colocarse entre el router (con un único cable) y la red local (conectado al switch o al hub de la LAN)

Sea el tipo de firewall que sea, generalmente no tendrá mas que un conjunto de reglas en las que se examina el origen y destino de los paquetes del protocolo TCP/IP. En cuanto a

[20] http://www.pello.info/filez/firewall/iptables.html
[21] http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=14

protocolos es probable que sean capaces de filtrar muchos tipos de ellos, no solo los TCP, también los UDP, los ICMP y otros protocolos vinculados a VPNS. Este podría ser (en pseudo-lenguaje) el conjunto de reglas de un firewall del segundo gráfico:

Politica por defecto ACEPTAR.
Todo lo que venga de la red local al firewall ACEPTAR
Todo lo que venga de la ip de mi casa al puerto tcp 22 ACEPTAR
Todo lo que venga de la ip de casa del jefe al puerto tcp 1723 ACEPTAR
Todo lo que venga de hora. rediris.es al puerto udo 123 ACEPTAR
Todo lo que venga de la red local y vaya al exterior ENMASCARAR
Todo lo que venga del exterior al puerto tcp 1 al 1024 DENEGAR
Todo lo que venga del exterior al puerto tcp 3389 DENEGAR
Todo lo que venga del exterior al puerto udp 1 al 1024 DENEGAR

En definitiva lo que se hace es:
- Habilita el acceso a puertos de administración a determinadas IPs privilegiadas
- Enmascara el trafico de la red local hacia el exterior (NAT, una petición de un pc de la LAN sale al exterior con la ip pública), para poder salir a internet
- Deniega el acceso desde el exterior a puertos de administración y a todo lo que este entre 1 y 1024.

Hay dos maneras de implementar un firewall:
1) Política por defecto ACEPTAR: en principio todo lo que entra y sale por el firewall se acepta y solo se denegará lo que se diga explícitamente.
2) Política por defecto DENEGAR: todo esta denegado, y solo se permitirá pasar por el firewall aquellos que se permita explícitamente.

Como es obvio imaginar, la primera política facilita mucho la gestión del firewall, ya que simplemente nos tenemos que preocupar de proteger aquellos puertos o direcciones que sabemos que nos interesa; el resto no importa tanto y se deja pasar. Por ejemplo, si queremos proteger una máquina linux, podemos hacer un netstat -ln (o netstat –an), saber que puertos están abiertos, poner reglas para proteger esos puertos y ya está. ¿Para qué vamos a proteger un puerto que realmente nunca se va a abrir?

El único problema que podemos tener es que no controlemos que es lo que esta abierto, o que en un momento dado se instale un software nuevo que abra un puerto determinado, o que no sepamos que determinados paquetes ICMP son peligrosos. Si la política por defecto es ACEPTAR y no se protege explícitamente, nos la estamos jugando un poco.

En cambio, si la política por defecto es DENEGAR, a no ser que lo permitamos explícitamente, el firewall se convierte en un auténtico MURO infranqueable. El problema es que es mucho más difícil preparar un firewall así, y hay que tener muy claro como funciona el sistema (sea iptables o el que sea) y que es lo que se tiene que abrir sin caer en la tentación de empezar a meter reglas super-permisivas.
Esta configuración de firewall es la recomendada, aunque no es aconsejable usarla si no se domina mínimamente el sistema.

Importante
El orden en el que se ponen las reglas de firewall es determinante. Normalmente cuando hay que decidir que se hace con un paquete se va comparando con cada regla del firewall hasta que se encuentra una que le afecta (match), y se hace lo que dicte esta regla (aceptar o denegar); después de eso NO SE MIRARÁN MÁS REGLAS para ese paquete. El peligro es que si ponemos reglas muy permisivas entre las primeras del firewall, puede que las siguientes no se apliquen y no sirvan de nada.

5.5.1 IPtables
IPtables es un sistema de firewall vinculado al kernel de linux que se ha extendido enormemente a partir del kernel 2.4 de este sistema operativo. Al igual que el anterior sistema ipchains, un firewall de iptables no es como un servidor que lo iniciamos o detenemos o que se pueda caer por un error de programación; iptables esta integrado con el kernel, es parte del sistema operativo. Para ponerlo en marcha lo que se hace es aplicar reglas. Para ello se ejecuta la instrucción iptables, con el que añadimos, borramos, o creamos reglas."[22]
"Su funcionamiento es simple: a iptables se le proporcionan unas reglas, especificando cada una de ellas unas determinadas características que debe cumplir un paquete. Además, se

[22] http://www.pello.info/filez/firewall/iptables.html

especifica para esa regla una acción o target. Las reglas tienen un orden, y cuando se recibe o se envía un paquete, las reglas se recorren en orden hasta que las condiciones que pide una de ellas se cumplen en el paquete, y la regla se activa realizando sobre el paquete la acción que le haya sido especificada.

Estas acciones se plasman en los que se denominan **targets**, que indican lo que se debe hacer con el paquete. Los más usados son bastante explícitos: **ACCEPT**, **DROP** y **REJECT**, pero también hay otros que nos permiten funcionalidades añadidas y algunas veces interesantes: LOG, MIRROR.

En cuanto a los paquetes, el total del sistema de filtrado de paquetes del kernel se divide en tres tablas, cada una con varias **chains** a las que puede pertenecer un paquete, de la siguiente manera.

- **filter**: Tabla por defecto, para los paquetes que se refieran a nuestra máquina
  - **INPUT**: Paquetes recibidos para nuestro sistema
  - **FORWARD**: Paquetes enrutados a través de nuestro sistema
  - **OUTPUT**: Paquetes generados en nuestro sistema y que son enviados
- **nat**: Tabla referida a los paquetes enrutados en un sistema con Masquerading
  - **PREROUTING**: Para alterar los paquetes según entren
  - **OUTPUT**: Para alterar paquetes generados localmente antes de enrutar
  - **POSTROUTING**: Para alterar los paquetes cuando están a punto para salir
- **mangle**: Alteraciones más especiales de paquetes
  - **PREROUTING**: Para alterar los paquetes entrantes antes de enrutar
  - **OUTPUT**: Para alterar los paquetes generados localmente antes de enlutar.[23]"

5.5.1.1 Elementos básicos

-Ordenes básicas:

iptables –F : efectivamente, flush de reglas

iptables –L : si, listado de reglas que se están aplicando

[23] http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=14

iptables –A : append, añadir regla

iptables –D : borrar una reglas, etc.

5.5.1.2 Ejemplo de regla:

#Regla que acepta conexiones al puerto 80

iptables -A INPUT -i eth0 -s 0.0.0.0/0 -p TCP --dport www -j ACCEPT

5.5.1.3 Anatomia de la Regla:

iptables: instrucción iptables (no hay que olvidar que las reglas son un shell script)

-A: append, opción para añadir la regla

INPUT: estado del paquete (al entrar es input).

-i eth0: interfaz de red eth0

-s 0.0.0.0/0: dirección de acceso (cualquiera en este caso)

-p TCP: tipo de puerto

--dport: puerto de destino

-j ACCEPT: destino del paquete (se acepta, podría ser DROP, LOG, REJECT,..)

5.5.1.4 Guía rápida de flags:

-s : source address. Ej: -s 192.168.1.0/24

-d : destino. Ej: -d 84.56.73.3

-p : tipo de protocolo(TCP,UDP,ICMP). Ej: -p TCP

--sport : puerto de origen

--dport: puerto de destino

-i = –in-interface : el interfaz por el que se entra (eth0,eth1, ppp0,…)

-o = --out-interface: el interfaz por el que se sale (eth0,eth1, ppp0,…)

-Notas:

-i se usa con reglas INPUT y FORWARD

-o se usa con reglas FORWARD y OUTPUT

Algunas de las opciones que se permiten en las instrucciones de arriba son:

- **`-v`**: Modo verboso, útil sobre todo con `iptables -L`.
- **`-n`**: las direcciones IP y números de puertos se mostrarán numéricamente (sin resolver nombres).

`--line-numbers`: Muestra los números de regla de cada regla, de manera que sea más fácil identificarlas para realizar operaciones de inserción, borrado.

5.5.1.5 Firewall con Iptables

- "Para crear nuestro sencillo firewall doméstico, tendremos primero que preguntarnos qué es lo que deseamos que haga. Lo más usual, en un equipo que se usa para conexiones a Internet de manera normal (no es servidor de nada, etc.) es que deseemos de nuestro firewall lo siguiente:

  - Que nos permita realizar conexiones TCP hacia afuera de nuestra máquina (si no, no podríamos hacer casi nada).
  - Que no permita realizar conexiones TCP desde afuera hacia nuestra máquina, para evitar que alguien intente conectarse a nuestros servidores web, ftp, telnet, X.
  - Que permita el tráfico de paquetes TCP (paquetes que no establezcan conexiones) en ambas direcciones, pues necesitamos tráfico bidireccional de paquetes al usar casi cualquier cosa en Internet.
  - Que prohíba el tráfico UDP desde afuera de nuestra máquina, a excepción del necesario para las respuestas por parte de nuestros servidores DNS, que provendrán de su puerto UDP 53.
  - En caso de tener una intranet, que no aplique estas restricciones al tráfico proveniente de y enviado hacia la intranet, ya que en esta red interna probablemente sí nos interese poder acceder remotamente a nuestra máquina."[24]

  Creación del firewall

  "Para crear nuestro firewall, iremos introduciendo una a una las reglas que necesitamos:

  Primera regla: permitiremos cualquier tráfico que provenga de nuestro interfaz de

[24] http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=14

loopback (`lo`), para ello insertaremos en el chain INPUT (que se encarga de los paquetes que llegan con destino a nuestra máquina), de la tabla **filter** la siguiente regla:

```
$ /sbin/iptables -A INPUT -i lo -j ACCEPT
```

Atención: es importante aquí **respetar las mayúsculas**, pues los nombres del chain y del target son INPUT Y ACCEPT, no input o accept.

Segunda regla: si disponemos de intranet, permitiremos todo el tráfico que provenga de nuestro interfaz de red interna. Por ejemplo, imaginando que tuviésemos una ethernet en el interfaz `eth0`, haríamos:

```
$ /sbin/iptables -A INPUT -i eth0 -j ACCEPT
```

El hecho de que omitamos la dirección de origen, de destino... implica que nos referimos a todas.

Tercera regla: impediremos el paso de cualquier paquete TCP proveniente del exterior que intente establecer una conexión con nuestro equipo. Estos paquetes se reconocen por tener el flag SYN asertado y los flags ACK y FIN desacertados. Para decirle a la regla que reconozca específicamente estos paquetes, usaremos una opción que se puede usar cuando el protocolo del paquete es declarado como **tcp**, la opción **--syn**. De la siguiente manera:

```
$ /sbin/iptables -A INPUT -p tcp --syn -j REJECT --reject-with icmp-
port-unreachable
```

Y vemos también el uso de una opción del target **REJECT**, que nos permite elegir de qué manera debe ser rechazado el paquete. Posibles valores son icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable, icmp-proto-unreachable, icmp-net-prohibited y icmp-host-prohibited.

Cuarta regla: Antes de declarar que deseamos prohibir cualquier tráfico UDP hacia nuestra máquina, y dado que las reglas se recorren en orden hasta que una de ellas se activa con el paquete**,** tendremos que añadir ahora una regla que nos permita recibir las respuestas de nuestro/s servidor/es DNS cuando nuestro sistema les realice alguna

consulta. Estas respuestas, vía UDP, saldrán del puerto 53 del servidor DNS. La regla, pues, será:

```
$ /sbin/iptables -A INPUT -p udp --source-port 53 -j ACCEPT
```

Donde **--source-port** es una opción presente cuando el protocolo es **udp** (también cuando es **tcp**) y nos permite en este caso especificar que la consulta provenga del puerto destinado al DNS.

Quinta regla: Prohibimos ahora el resto del tráfico UDP. La regla de por sí implica a todo el tráfico UDP, pero como un paquete sólo activará esta regla si no ha activado la anterior, los paquetes UDP referentes a una transacción con un servidor de nombres no se verán afectados.

```
$ /sbin/iptables -A INPUT -p udp -j REJECT --reject-with icmp-port-
unreachable
```

Dado que los targets por defecto (denominados policy o política) en la tabla filter son ACCEPT, si un paquete no activa ninguna de las reglas, será aceptado, de manera que no tendremos que preocuparnos de, por ejemplo, los paquetes de tráfico normal de TCP, ya que estos serán aceptados al no activar regla alguna.
Si ahora escribimos:

```
$ /sbin/iptables -L -v
```

Deberíamos obtener algo como:

Chain INPUT (policy ACCEPT 3444 packets, 1549K bytes)
pkts bytes target prot opt in out source destination
11312 3413K ACCEPT all -- lo any anywhere anywhere
0 0 ACCEPT all -- eth0 any anywhere anywhere
0 0 REJECT tcp -- any any anywhere anywhere tcp flags:SYN,RST,ACK/SYN reject-with icmp-port-unreachable
0 0 ACCEPT udp -- any any anywhere anywhere udp spt:domain
0 0 REJECT udp -- any any anywhere anywhere reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 15046 packets, 4218K bytes)
pkts bytes target prot opt in out source destination

De modo que nuestro pequeño y básico firewall doméstico ya está configurado. Nuestro equipo es ahora muchísimo más seguro, puesto que los ataques a nuestro sistema requerirían ahora mucha más elaboración, tiempo y esfuerzo por parte del atacante, de manera que nuestra condición de insignificantes ya empezará a ser importante como garante de seguridad.

5.5.1.6 Guardar y rehusar nuestra configuración de iptables

Pero, si una vez realizadas estas configuraciones, apagásemos nuestro equipo, todo esto se perdería, y tendríamos que volver a realizar una a una las sentencias de configuración.
Para evitar esto, `iptables` cuenta con dos programas auxiliares: **`iptables-save`** e **`iptables-restore`**, el primero de los cuales nos permite sacar por salida estándar el contenido de nuestras tablas IP, y el seguno nos permite, a partir de la salida generada por iptables-save recuperar la configuración de las tablas.

De manera que para volcar la configuración de nuestro firewall en un fichero ejecutaremos:

$ /sbin/iptables-save -c > [fichero]

Donde **-c** es una opción que nos permite guardar los contadores del número de paquetes que activaron cada regla.
Y, cuando queramos, podremos recuperar la configuración del firewall con:

$ /sbin/iptables-restore -c < [fichero]

En cuyo caso –c tiene el mismo significado que con iptables-save.

Estas llamadas a iptables-save e ipatbles-restore podrán ser incluidas en los scripts adecuados para que se lleven a cabo de manera automática en el arranque y el cierre del sistema."[25]

- Para crear un script de un firewall básico utilizando iptables

"Se generan las reglas en cualquier editor de texto guardándolo con el nombre que gusten, después se cambia los permisos para que se pueda ejecutar:

#> chmod 700 fw_equipo
y después ejecútalo:
#> ./fw_equipo
Si no manda errores, listo, tu firewall esta protegiéndote de ataques y de accesos indeseados.

```
# ------------------------------------------------------------------------
# firewall para un solo equipo conectado a través de MODEM.
# (1) se eliminan reglas previas que hubiera y cadenas definidas por el usuario
iptables -F
iptables -X

# (2) se establecen políticas "duras" por defecto, es decir solo lo que se autorice
# explícitamente podrá ingresar o salir del equipo
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# (3)a la interface local (localhost) se le permite todo
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# (4) evitamos ataques syn-flood limitando el acceso de paquetes nuevos
```

[25] http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=14

```
# desde internet a solo 4 por segundo y los demás se descartan
iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 4 -j DROP

# (5) se evitan paquetes tcp que sean nuevos y que no tengan el flag SYN
# es decir, hay ataques o escaneos que llegan como conexiones nuevas
# pero sin ser paquetes syn, definitivamente no nos interesan.
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

# (6) todo lo que sea icmp (ping) y que intente entrar, se descarta
# con esto bloqueamos cualquier tipo de paquetes con protocolo icmp
# evitando ataques como el del ping de la muerte, aunque esta regla
# podría provocar problemas de comunicación con algunos ISP.
iptables -A INPUT -p icmp -j DROP

# (7) por último las dos siguientes reglas permiten salir del equipo
# (output) conexiones nuevas que nosotros solicitamos, conexiones establecidas
# y conexiones relacionadas, y deja entrar (input) solo conexiones establecidas
# y relacionadas.
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
# ----------------------------------------------------------------------
```

El paso (1) hace un borrado de las reglas que ya hubiera, muchas distribuciones se instalan con un juego previo de reglas de iptables que para este caso no nos sirven, por eso las eliminamos, así como la opción X que permite eliminar las cadenas de reglas personalizadas. Si quieres ver que reglas tienes actualmente en tu firewall usa la opción –L.

```
#> iptables -L
```

En el segundo paso (2) establecemos las políticas del firewall, hay dos tipos de políticas ACCEPT y DROP, en el primer caso, estaríamos aceptando TODO lo que entre y salga del equipo y después se tendría que negar lo que no se quiera, cosa bastante tediosa e insegura. Lo mejor en firewalls es siempre establecer políticas DROP, con esto por

default, todo, absolutamente todo es prohibido de entrar o salir, asi que tenemos que ir poniendo reglas que abran los puertos o conexiones que si queramos. La desventaja de crear un firewall con políticas DROP es que suelen ser más complejos de crear y mantener, pero valen la pena.

Se establecen entonces para los paquetes que entran al equipo INPUT, para los paquetes que salen del equipo OUTPUT y para paquetes que atraviesan el equipo FORWARD, aunque esta última política en el caso de una PC con una sola tarjeta de red es innecesaria, o más bien está sobrando.
El paso (3) es necesario e importante ya que como estamos negando todo por default eso abarca también a nuestro dispositivo de red local o virtual, localhost, y varios servicios que trabajan de manera local en nuestro equipo como el sistema de ventanas X, hacen uso de este dispositivo para trabajar. Localhost no se conecta de ninguna manera al exterior a la LAN o Internet, por lo que es seguro simplemente decirle al firewall que acepte todo de entrada y salida que provenga de localhost.

Los pasos (4), (5) y (6) podrían parecer redundantes o innecesarios ya que como se observa su target es DROP lo que quiere decir que si el paquete se cumple en alguna de esas tres reglas se descartará. Realmente no sobran, añaden un extra de seguridad al firewall, ya que recordemos que las reglas son chequeadas contra los paquetes que entran o salen en un orden estrictamente secuencial a como las introducimos. Las dos últimas reglas permitirán el paso de paquetes (tal vez), así que lo que hacemos es depurar los paquetes en los pasos (4), (5) y (6) para que llegan un poco mas controlados a la decisión del paso (7) si son aceptados o no.
Finalmente en el paso (7) se permiten un par de reglas que forman un firewall de estado completo, donde se analiza el estado de los paquetes, sin importar su protocolo, destino, etc. En la regla de INPUT es decir lo que entra al equipo desde Internet solo se permiten paquetes cuyo estado ya este registrado en la tabla de conexiones del kernel. Esto implica que ningún paquete que sea nuevo será permitido, solo se aceptará lo que previamente se haya solicitado desde nuestro equipo y que es precisamente la última regla, la de OUTPUT, donde establecemos que pueden salir paquetes nuevas, como

una petición de página web, que son paquetes NEW, paquetes previamente establecidos (ESTABLISHED), como comunicaciones de chat y conexiones relacionadas a una establecida (RELATED), por ejemplo conexiones ftp que establecen conexión de control y de datos en los puertos 0 y 21."[26]

5.6 Cifrado del disco duro

5.6.1 CFS (Sistema de Ficheros Criptográfico)

"El CFS te permite guardar los datos en el disco duro en formato cifrado, y es significativamente más sencillo de utilizar que un programa de cifrado de ficheros (como PGP) si lo que se quiere es almacenar muchos ficheros y directorios de los que se quiere tener alejados a los curiosos. El sitio oficial de distribución está en: http://www.cryptography.org/, y también se encuentran disponibles RPM's en: ftp://ftp.replay.com/pub/replay/linux/redhat/, y los ficheros binarios Debian están en: http://www.debian.org/Packages/stable/otherosfs/cfs.html.

5.6.2 TCFS

El TCFS es una utilidad de cifrado de datos a nivel del kernel, similar al CFS. Sin embargo tiene algunas ventajas sobre el CFS; puesto que está implementado en el kernel, es significativamente más rápido. Está fuertemente arraigado con el NFS, lo cual quiere decir que se pueden servir datos de forma segura en una máquina local, o a través de la red. Descifra datos en la máquina cliente, de modo que cuando se utiliza sobre la red, nunca se transmite la contraseña. La única pega es que todavía no se ha adaptado a la serie 2.2 del kernel. El TCFS se puede conseguir en: http://tcfs.dia.unisa.it/

5.6.3 PPDD

El PPDD permite crear una partición de disco cifrada, puede ser una partición real o un dispositivo loopback (el cual reside en un fichero, pero se monta como un sistema de

[26] http://www.linuxtotal.com.mx/fw_basico.html

ficheros). Utiliza el algoritmo blowfish, el cual es relativamente rápido y está probado. El PPDD se puede conseguir desde: http://linux01.gwdg.de/~alatham/.

5.6.4 Directorio Raíz Cifrado

El Encrypted Home Directory, Directorio Raíz Cifrado, funciona de forma parecida al CFS, sin embargo está orientado a proporcionar un sólo directorio cifrado. Resumiendo, crear un fichero de tamaño X en /crypt/ con tu UID, y lo monta en un dispositivo loopback de forma que se pueda acceder a él. El truco está en que los datos se cifran y descifran al vuelo, cuando se accede a él (como con el CFS). La única pega es que el software todavía está en desarrollo, así que haz copia de seguridad de cualquier dato importante. Se puede descargar desde: http://members.home.net/id-est/."[27]

5.6 Resumen

En este capitulo se presentan herramientas de monitoreo que nos ayudarán a verificar posibles debilidades de un servidor al igual que se controlará el trafico de la red y se monitorizará el flujo de tramas, adicionalmente podremos aprender a crear un firewall para contrarrestar los ataques.

[27] http://es.tldp.org/Manuales-LuCAS/GSAL/gsal-19991128-htm/cifrado.htm

CAPITULO 6
PRACTICA

6.1 Ejemplos prácticos.

- **Uso de Tripwire**
  - **Descripción**

    Tripwire monitorea rutinariamente la integridad de una gran cantidad de archivos que tienden a ser blanco de los atacantes.
  - **Instrucciones usadas**

    Intrucción:

    tripwire –m c

    Salida/Explicación

    Realiza un chequeo comparando con la base de datos y emite un reporte nos indica los archivos revisados, eliminados, modificados y actualizados.

    Intrucción:

    twprint -m r --twrfile /var/lib/tripwire/report/localhost.localdomain-20060305-165902.twr |less

    Salida/Explicación

    Muestra el reporte de una revisión realizada.

- **Uso de Nmap**
  - **Descripción**

    Nmap es una herramienta para el escaneo de puertos probablemente de las mejores que existen siendo software libre.
  - **Instrucciones usadas**

    Intrucción:

    nmap 192.168.55.2

    Salida

    [root@localhost]# nmap 192.168.55.2

    Starting Nmap 4.01 (http://www.insecure.org/nmap/) at 2006-03-05 15:33 ECT

Interesting ports on 192.168.55.2:

(The 1666 ports scanned but not shown below are in state: closed)

PORT STATE SERVICE

7/tcp open echo

22/tcp open ssh

111/tcp open rpcbind

1016/tcp open unknown

1241/tcp open nessus

3306/tcp open mysql

Nmap finished: 1 IP address (1 host up) scanned in 13.091 seconds

Intrucción:

nmap -sU localhost

Salida

[root@localhost]# nmap -sU localhost

Starting Nmap 4.01 (http://www.insecure.org/nmap/) at 2006-03-05 15:39 ECT

Interesting ports on localhost.localdomain (127.0.0.1):

(The 1478 ports scanned but not shown below are in state: closed)

PORT STATE SERVICE

111/udp open|filtered rpcbind

631/udp open|filtered unknown

1010/udp open|filtered unknown

1013/udp open|filtered unknown

Nmap finished: 1 IP address (1 host up) scanned in 1.238 seconds

Intrucción:

nmap -O 192.168.55.2

Salida

[root@localhost]# nmap -O 192.168.55.2

Starting Nmap 4.01 (http://www.insecure.org/nmap/) at 2006-03-05 15:42 ECT

Interesting ports on 192.168.55.2:

(The 1666 ports scanned but not shown below are in state: closed)

PORT STATE SERVICE

7/tcp open echo

22/tcp open ssh

111/tcp open rpcbind

1016/tcp open unknown

1241/tcp open nessus

3306/tcp open mysql

Device type: general purpose

Running: Linux 2.4.X|2.5.X|2.6.X

OS details: Linux 2.5.25 - 2.6.8 or Gentoo 1.2 Linux 2.4.19 rc1-rc7, Linux 2.6.3 - 2.6.10

Uptime 0.077 days (since Sun Mar 5 13:52:52 2006)

Nmap finished: 1 IP address (1 host up) scanned in 15.092 seconds

Intrucción:

nmap -sP 192.168.55.2/24

Salida

[root@localhost]# nmap -sP 192.168.55.2/24

Starting Nmap 4.01 (http://www.insecure.org/nmap/) at 2006-03-05 15:43 ECT

Host 192.168.55.2 appears to be up.

Host 192.168.55.3 appears to be up.

MAC Address: 00:03:47:88:DE:1F (Intel)

Nmap finished: 256 IP addresses (2 hosts up) scanned in 31.155 seconds

Intrucción:

-sS 192.168.55.2 -p 80,110

Salida

[root@localhost nessus]# nmap -sS 192.168.55.2 -p 7,22

Starting Nmap 4.01 (http://www.insecure.org/nmap/) at 2006-03-05 15:47 ECT

Interesting ports on 192.168.55.2:

PORT STATE SERVICE

7/tcp open echo

22/tcp open ssh

Nmap finished: 1 IP address (1 host up) scanned in 13.012 seconds

- **Uso de Snort**
  - **Descripción**

Snort es un detector de intrusiones de red capaz de realizar análisis del tráfico de red en tiempo real e indicarnos posibles anomalías.

Instrucción:

snort -vde

Salida

Imprime los encabezados de los paquetes y los datos, además los encabezados de la capa de enlace.

Instrucción:

snort -dev -l ./log

Salida

Lo que hemos hecho es agregar el switch -l indicándole que guarde los datos en el directorio ./log, si este directorio no existe hay que crearlo primeramente.

Instrucción:

snort -d -h 192.168.55.0/24 -l /var/log/snort -c /etc/snort/snort.conf

Salida

Procederá a guardar los eventos inusuales dentro de /var/log/snort y leerá la configuración de estos eventos en: /etc/snort/snort.conf

Instrucción:

snort -d -h 192.168.1.0/24 -l /var/log/snort -c /etc/snort.conf -D

Salida

Si queremos correr snort en modo de demonio, agregamos el switch -D:

- **Uso de IPtraf**
  - **Descripción**

    Esta es una herramienta que permite el análisis de tráfico y redes.

    Instrucción:

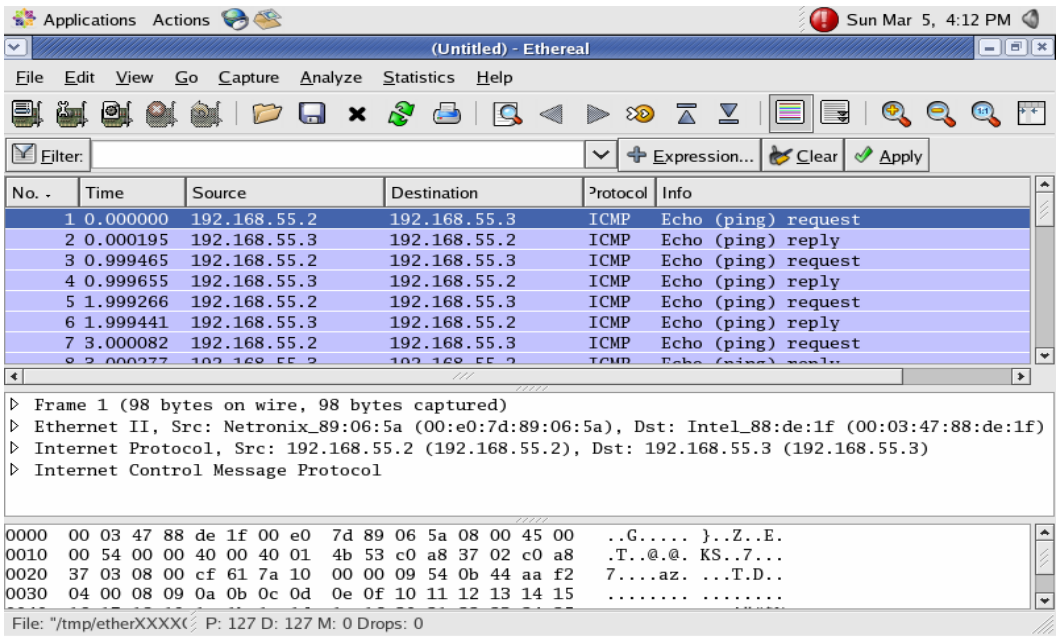
    iptraf

    Salida

    La salida de esta instrucción nos muestra una GUI (interfase gráfica usuario) que nos permite visualizar el tráfico de la red.

- **Uso de Ethereal**
  - **Descripción**

    Es un capturador de paquetes en tiempo real muy poderoso, el cual le puede permitir al administrador de sistemas diagnosticar problemas a bajo nivel examinando los datos que viajan por la red. Se accede al menú aplicaciones y la opción Ethereal Network Analyzer y saldrá la siguiente pantalla.



- **Uso de Iptables**
  - **Descripción**

    iptables es la herramienta que nos permite configurar las reglas del sistema de filtrado de paquetes del kernel de Linux, desde su versión 2.4. Con esta herramienta, podremos crearnos un firewall adaptado a nuestras necesidades.

    Instrucción:

    #----------

    iptables -F

    iptables –X

    iptables -P INPUT DROP

    iptables -P OUTPUT DROP

    iptables -P FORWARD DROP

    iptables -A INPUT -i lo -j ACCEPT

    iptables -A OUTPUT -o lo -j ACCEPT

    iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 4 -j DROP

    iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

    iptables -A INPUT -p icmp -j DROP

    iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

    iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

    # ----------

    Salida

    Primeramente realizamos un borrado de las reglas que ya hubiera Si quieres ver que reglas tienes actualmente en tu firewall usa:

    #> iptables -L

Luego establecemos las políticas del firewall, que son: ACCEPT o DROP.
Se procede habilitar el localhost. Si el paquete se cumple en alguna de las reglas de Drop se descartará.
Las dos últimas reglas permitirán el paso de paquetes (tal vez), así que lo que hacemos es depurar los paquetes en los pasos siguientes para que lleguen un poco más controlados a la decisión del último paso si son aceptados o no.
Finalmente en el último paso se permiten un par de reglas que forman un firewall de estado completo, donde se analiza el estado de los paquetes, sin importar su protocolo, destino, etc.

CONCLUSIONES

Al culminar la presente monografía hemos llegado a las siguientes conclusiones:

- Que Linux Centos V. 4.2 es un Sistema Operativo que se encuentra disponible debido a su código abierto y que tiene las mismas funcionalidades que la ultima versión de Linux Redhat y adicionalmente se encuentra al día en las actualizaciones por lo tanto convirtiéndose en un sistema robusto en lo que se refiere a seguridades.

- El administrador o usuario de Linux debe analizar los servicios que tenga su sistema pero al mismo tiempo debe medir la seguridad de este, debido a que entre más seguro su sistema tendrá menos funcionalidades y si tiene más funcionalidad estaría propenso a tener problemas de seguridad.

- Las herramientas observadas en la presente monografía son de gran utilidad para controlar, monitorizar y prevenir posibles fallas de seguridad que impliquen riesgos en su sistema, igualmente son opensource y fáciles de utilizar.

- Al haber concluido esta monografía nos sentimos satisfechas de haber puesto en práctica los conocimientos adquiridos en el curso de graduación y esperamos que esta monografía sirva de ayuda a las personas interesadas en mantener seguro su Sistema Linux Centos V. 4.2.

GLOSARIO

Bios: (Basic Input Output System, Sistema de Entrada /Salida Básico): Es una memoria ROM, EPROM, Oflasy-Ram la cual contiene las rutinas de más bajo nivel. que hace posible que el ordenador pueda arrancar.

Cracker: Es una persona que intenta acceder a un sistema informático sin autorización.

Ethernet: Ethernet es el nombre de una tecnología de redes de computadores de Área Local.

Intranet: Es una infraestructura basados en los estándares y tecnologías de Internet que soporta el compartir información dentro de un grupo bien definido y limitado.

kernel: El Kernel o mejor conocido como núcleo es la parte fundamental de un Sistema Operativo.

LAN: (Local Área Network, Red de Área Local) Se refiere a las redes locales de ordenadores.

Libpcap: Es una librería que proporciona funciones para la captura de paquetes a nivel de usuario, utilizada en la monitorización de redes.

MAC: Es un identificador físico, de 48 bits almacenados dentro de una tarjeta de Red.

Plugins: Realizan un combinación de acciones o tareas encadenadas de modo automático en un software.

Tcpdump: El programa Tcdump te permite monitorear tráfico de red en tiempo real.

BIBLIOGRAFIA

CRIPTONOMICON. Seguridad en Linux. http://www.iec.csic.es/criptonomicon/linux/introsegnucleo.html [consulta 9 de Diciembre del 2005].

ELRINCONDEWINDOWS. Sistema de Archivos. http://quicksitebuilder.cnet.com/camiloalie/ElRinconDeWindows/id124.html [consulta 10 de Enero del 2006].

GEOCITIES.COM. Seguridad Linux Como. http://www.geocities.com/galinux2000/numero1/doc-es/seguridadelinuxcomo.htm [consulta 10 de Enero del 2006]

SEGURIDAD BASICA EN LINUX. Seguridad y Passwords. http://david.f.v.free.fr/ponencias/Seguridad_Basica_en_Linux/seguridad-3.html [consulta 12 de Enero del 2005]

TUTORIALES. Guía Breve de Tripwire. http://es.tldp.org/Tutoriales/GUIA_TRIPWIRE/guia_tripwire.html [consulta 19 de Enero del 2006].

PELLO.INFO. Iptables Manual Práctico. http://www.pello.info/filez/firewall/iptables.html [consulta 5 de Febrero del 2006]

LINUX, ANTIVIRUS, ANTISPAM, CURSOS. Introducción al Nmap. http://www.ernestoperez.com/documentacion/introduccion_al_nmap [consulta 6 de Febrero del 2006]

LINUX, ANTIVIRUS, ANTISPAM, CURSOS. Instalación de un NIDS – Snort. http://www.ernestoperez.com/documentacion/instalacion_de_un_nids_snort.html [consulta 5 de Febrero del 2006]

EL RINCON DEL PROGRAMADOR. Configuración de Firewall en Linux con Iptables. http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=14 [consulta 7 de Febrero del 2006]

LINUX ALBACETE. Firewall. http://linux.ideando.net/articulos/hs/hs11.htm [consulta 7 de Febrero del 2005]

LINUXTOTAL.COM.MX. Firewall Básico. http://www.linuxtotal.com.mx/fw_basico.html [consulta 7 de Febrero del 2006]

MANUALES-LUCAS. Cifrado de Servicios/Datos. http://es.tldp.org/Manuales-LuCAS/GSAL/gsal-19991128-htm/cifrado.htm [consulta 16 de Febrero del 2006]