01132
61

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"Implementando una red segura con software libre"

T E S I S

QUE PARA OBTENER EL TÍTULO DE

Ingeniero en Computación

P R E S E N T A :

FERNANDO JAVIER MARTINEZ MENDOZA

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.
NOMBRE: Fdo. Javier Martínez Mendoza
FECHA: 24-Jun-2003
FIRMA:

DIRECTORA DE TESIS:
ING. LAURA SANDOVAL MONTAÑO

2003

TESIS CON
FALLA DE ORIGEN

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PAGINACIÓN DISCONTINUA

A mi Abuelita,

el apoyo e impulso más grande de mi vida

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

Agradecimientos

Agradezco a Dios por ayudarme en todo este tiempo y darme la inteligencia y fuerza para estudiar y progresar.

A mi familia por siempre estar conmigo y ser tan importantes para mi. Agradezco a mi Abue por siempre orientarme y ayudarme, a mi Mamá y a mis hermanos Daniel, Pork y Mayo por darme la oportunidad de ser parte de sus vidas. A mi Moye por enseñarme lo que es vivir con ganas y a ser agradecido, a Enrique por siempre estar con nosotros, hacernos la vida tan fácil y tener listo siempre un buen consejo. A mis tíos Nena, Rober, Pepé, Carmen y Erikita por siempre hacerme sonreír y saber dirigirme en la vida.

A Keka por todo este tiempo, por apoyarme y ayudarme en los momentos buenos y malos. Gracias por todo tu cariño.

A las personas que hacen más agradable mi vida y me acompañan e impulsan, siempre divirtiéndome y sacándome de mi rutina. Entre estas a mis amigos de siempre: Ricky Mouse, Iván, Hetorín, Laura, Rafa, Erick, Alex, Chivis y Tanita.

A la UNAM y muy especialmente a la Facultad de Ingeniería. A mi directora de tesis Ing. Laura Sandoval y a todos los profesores que compartieron conmigo un poco de su conocimiento. A la UNICA y toda la gente grande que me ayudo ahí, por ser el inicio de toda esta tesis y por darme la oportunidad de dar mis primeros pasos en este mundo de la seguridad y la administración.

Al Instituto de Física por el apoyo recibido, muy especialmente a: Dr. Guillermo Ramírez, Dr. Octavio Miramontes y Dr. Matías Moreno. Gracias por tanto impulso y por la oportunidad de desarrollarme.

A mis compañeros que siempre me ayudaron cuando lo necesité, además de que me hacen pasar tan bueno momentos: Natorro, Neptali, Alex Juárez, Carlos Castaños, Chova, Chayo, Israel, Juan, Oscar, Mariana, Laura, Gaby, Ricardo, Mago, Rafa Sandoval, Noé Cruz, Yesenia, Anny, Erika, Adrián, Filip, Sonnii, etc, etc, etc.

A todas las personas que por falta de espacio (y de memoria) no incluyo, pero que no dejan de ser importantes en mi vida (y también a los que faltan por llegar).

Índice general

Índice de figuras

Índice de cuadros

PREFACIO

TESIS CON
FALLA DE ORIGEN

Hoy vivimos con las computadoras y los servicios de redes como parte de nuestra vida cotidiana, tenemos que utilizar un servicio de red informática o alguna computadora para casi cualquier actividad, directa o indirectamente. Tal vez sin darse cuenta, todas las personas hacen trámites en el banco, en una oficina, para pagar el teléfono o simplemente al hacer una llamada telefónica en el que se utilizan una gran cantidad de servicios de cómputo y redes.

Las redes de computadoras han hecho que nuestra vida se vuelva mucho más fácil, pues ahora podemos realizar muchas tareas por medio de ellas sin la necesidad de salir de la casa o la oficina, y también se puede obtener mucha información de una manera rápida. Pero para poder explotar todos estos recursos necesitamos tener la seguridad de que nuestra información estará segura y que tendremos confidencialidad.

A la misma velocidad que han crecido las posibilidades con las redes de computadoras y las herramientas informáticas, han crecido las formas de hacer mal uso de todo esto. El mundo está lleno de personas que desean aprovecharse de cualquier oportunidad para poder realizar un fraude obteniendo información personal de los usuarios de estas redes, o pueden estar buscando hacer algún mal a otras personas o empresas por medio de estas herramientas.

Los administradores de redes y de sistemas saben que no se puede tener un sistema totalmente seguro, todos los días se descubren nuevos errores y huecos de seguridad, por lo que no se puede tener un servicio libre de los peligros que nacen día a día, esta tesis busca tener un acercamiento a la seguridad informática y de redes, explicará los principales problemas y algunas posibles soluciones; no busca ser una receta de cocina que se pueda seguir paso a paso para obtener la seguridad necesaria, pues se sabe que esta seguridad no existe, simplemente trata de mostrar un esquema de seguridad

en el que se puede tener cierto nivel de confianza en los servicios que se darán a través de nuestra red y nuestros equipos de cómputo. La seguridad en redes es un tema muy extenso y que puede encontrarse en una gran cantidad de libros tratando temas diferentes, esta tesis muestra desde un punto de vista básico qué es la seguridad en redes y la forma en que se utilizan herramientas de software libre para poder optimizar la seguridad de una red. Se utiliza software libre por la gran cantidad de beneficios que tiene, a lo largo de esta tesis se justifica cada uno de los programas de software libre que estamos utilizando.

El capítulo uno tratará de dar una introducción general al concepto de redes y la seguridad en ellas para poder hablar en capítulos posteriores de temas más avanzados teniéndo en cuenta de que el lector tiene un conocimiento básico del tema. En el segundo capítulo se hablará de la seguridad en una mayor profundidad, se definirá el concepto de seguridad y las tecnologías básicas existentes, además de las herramientas que se utilizarán tratando de explicar el porqué utilizamos esas herramientas y el beneficio que obtendremos al utilizarlas. En el tercer capítulo se hablará de un caso práctico: el Instituto de Física de la UNAM. Este sistema de seguridad se está empleando actualmente en el IFUNAM y mostraremos el gran número de problemas que ha resuelto. En el capítulo cuatro, se mostrará la instalación, configuración y uso de varias de las herramientas que se utilizaron para el sistema de seguridad mostrado por esta tesis. Por último se expondrán algunas conclusiones a las que se llego al desarrollar la investigación e instalación de la red que se plantea a lo largo de este trabajo.

Capítulo 1

Introducción

TESIS CON
FALLA DE ORIGEN

Las computadoras y los sistemas de comunicación electrónicos están llegando a ser parte de nuestras vidas, los encontramos en todos lados y para cualquier trámite que tengamos que efectuar tenemos que utilizar alguna de estas herramientas. Con el incremento de la computación en el comercio, el gobierno, la escuela y en la vida diaria viene la necesidad de proteger la información contenida en nuestros sistemas y prevenir intrusiones maliciosas o de cualquier otro tipo indeseable en nuestros sistemas. Para entender los temas que se tratan en esta tesis, debemos hablar primero de los conceptos de donde parte toda esta investigación: las redes de computadoras y la seguridad en ellas.

1.1. Redes de computadoras

La idea de tener redes de computadoras es tan vieja como las telecomunicaciones en sí. Considere a las personas que vivieron en la era de piedra, donde los tambores eran utilizados para transmitir mensajes entre las personas. Suponga que el hombre de las cavernas A quiere invitar al hombre de las cavernas B a un juego en el que se lanzan piedras uno a otro, pero B vive demasiado lejos para escuchar a A golpear su tambor. Entonces, ¿Cuáles son las opciones de A? El puede: 1) caminar al espacio de B, 2) conseguir un tambor más grande, o 3) pedir a C, que vive a la mitad del camino entre ellos, para que le pase el mensaje. Este último se conoce como trabajo en red (*networking*). Ahora ya no estamos en los tiempos de las cavernas, pero la idea es la misma. Tenemos millones de computadoras hablando con otras por

medio de cables, fibra óptica, microondas, rayos infrarrojos, ondas de radio, etc.

1.1.1. Concepto de red de computadoras

Una red consiste en dos o más computadoras unidas que comparten recursos y son capaces de realizar comunicaciones electrónicas. Las redes están unidas por cable, líneas de teléfono, ondas de radio, satélite, etc. Su objetivo principal es lograr que todos sus programas, datos y equipo estén disponibles para cualquiera de la red que lo solicite, sin importar la localización física del recurso y del usuario.

Componentes

Las redes de computadoras se forman con una serie de componentes de uso común y que en mayor o menor medida aparece siempre en cualquier instalación.

Servidores

Los servidores son equipos dedicados a alguna tarea en especial. Existen distintos tipos de servidores, como de correos, de web, de archivos, de passwords, de impresión, de direcciones, de nombres, etc., estos equipos al tener que responder a muchas peticiones, o al tener que realizar las tareas de otras máquinas, tienen que ser equipos poderosos. Un servidor de impresión se encargará de controlar el tráfico de la impresora ya que éste es el que accede a las demandas de las estaciones de trabajo y el que les proporcione los servicios que pidan las impresoras, archivos, Internet, etc. Es preciso contar con una computadora con capacidad de guardar información de forma muy rápida y de compartirla con la misma rapidez.

Estaciones de Trabajo

Son los equipos conectados al servidor. Las estaciones de trabajo no han de ser tan potentes como el servidor, simplemente necesita una tarjeta de red y el cableado, y software necesarios para comunicarse con el servidor.

Tarjeta de Red

La tarjeta de red o *NIC* es la que conecta físicamente a la computadora a la red. Las tarjetas de red más populares son por supuesto las tarjetas Ethernet, existen también conectores *Local Talk* así como tarjetas *TokenRing*.

Concentradores

Un concentrador o *hub* es un elemento que provee una conexión central para todos los cables de la red. Los *hubs* son cajas con un número determinado de conectores. Los *hubs* son concentradores que comparten el medio, cuando alguna terminal quiere enviar alguna señal al servidor, o enviar información a algún lugar de la red o Internet, el mensaje le llega a todos los que están conectados a este concentrador.

Distribuidores

Los distribuidores, conocidos como *switches*, son muy parecidos a los *hubs* pero a diferencia de los *hubs*, los *switches* son más "inteligentes". Los *switches* dividen el medio, es decir, cuando alguna estación de trabajo desea hablar con el servidor, o desea conectarse a alguna página en Internet, los switches deben enviar la señal sólo a la máquina a la que deben de enviarla, no enviarla a todas las otras estaciones de trabajo, o a servidores que no deben estar esperando ese tipo de llamadas.

Repetidores

Cuando una señal viaja a lo largo de un cable va perdiendo fuerza a medida que avanza. Esta pérdida de fuerza puede causar pérdida de información. Los repetidores amplifican la señal que reciben permitiendo así que la distancia entre dos puntos de la red sea mayor que la que un solo cable permite.

Puentes o *Bridges*

Los *bridges* se utilizan para segmentar redes grandes en redes más pequeñas, destinado a otra red pequeña diferente mientras que todo el tráfico interno seguirá en la misma red. Con esto se logra reducir el tráfico de la red.

Ruteador o *router*

Cuando tenemos una red conectada a Internet y mandamos alguna señal a una máquina que está fuera de nuestra red, necesitamos un equipo especial que nos diga qué camino seguir a través de la gran carretera que es la Internet. Un *router* es el equipo que nos envía por el camino que debe seguir nuestra conexión.

Gateway

El *gateway* es la puerta de enlace entre nuestra red y la red mundial. Ésta es la máquina por la que deben de pasar todas nuestras conexiones para poder ser enviadas a la Internet y también es por la que pasan todas las conexiones que vienen desde fuera de nuestra red.

Cableado de la red

El Cable es el medio a través del cual fluye la información por la red. Hay distintos tipos de cable de uso común en redes LAN. Una red puede utilizar uno o más tipos de cable, aunque el tipo de cable utilizado siempre estará sujeto a la topología de la red, el tipo de red que utiliza y el tamaño de ésta. Estos son los tipos de cable más utilizados en redes LAN:

Cable de par trenzado (*UTP Unshielded twisted pair*)
Cable de par trenzado con protección (*STP Shields twisted pair*)
Cable coaxial
Cable de fibra óptica

1.1.2. Modelos de comunicación de redes

Para satisfacer los requerimientos de los usuarios de cómputo remoto los grandes fabricantes han desarrollado una variedad de arquitecturas de comunicación. Algunas de estas arquitecturas definen y relacionan la forma en la que el hardware y el software deberán de comunicarse para poder tener la conexión entre diferentes tipos de tecnologías. Para esto se crearon distintos modelos de comunicación, a continuación se muestran los dos más utilizados:

Cuadro 1.1: Modelo OSI

Capa de aplicación
Capa de presentación
Capa de sesión
Capa de transporte
Capa de red
Capa de enlace
Capa Física

Modelo OSI

Con la finalidad de regularizar el desarrollo de productos para redes de comunicación de datos, fue elaborado un modelo abierto, que es llamado modelo OSI -*Open System Interconnection*- por la ISO -*International Organization for Standardization*-. En este modelo, el propósito de cada nivel es proveer servicios al nivel superior, liberándolo de los detalles de implementación de cada servicio. La información que se envía de un computador a otro debe pasar del nivel superior al nivel inferior atravesando todos los demás niveles de forma descendente, dentro del computador que origina los datos. Este modelo se dividió en siete capas que se muestran en el cuadro 1.1.

- Nivel Físico. Este nivel dirige la transmisión de flujos de bits, sin estructura aparente, sobre un medio de conexión. Se encuentra relacionado con condiciones eléctricas-ópticas, mecánicas y funcionales del interfaz al medio de transmisión. A su vez está encargado de aportar la señal empleada para la transmisión de los datos generados por los niveles superiores. En este nivel se define la forma de conectarse el cable a las tarjetas de red, cuántos pines debe tener cada conector y el uso funcional de cada uno de ellos. Define también la técnica de transmisión a emplear para el envío de los datos sobre el medio empleado. Se encarga de activar, mantener y desactivar un circuito físico. Este nivel trata la codificación y sincronización de los bits y es el responsable de hacer llegar los bits desde un computador a otro.

- Nivel de Enlace. Este nivel se encarga, en el computador de origen, de

alojar en una estructura lógica de agrupación de bits, llamada Trama (Frame), los datos provenientes de los niveles superiores. En el computador de destino, se encarga de agrupar los bits provenientes del nivel físico en tramas de datos (Frames) que serán entregadas al nivel de red. Este nivel es el responsable de garantizar la transferencia de tramas libres de errores de un computador a otro a través del nivel físico.

- Nivel de Red. Es responsable del direccionamiento de mensajes y de la conversión de las direcciones lógicas y nombres, en direcciones físicas. Está encargado también de determinar la ruta adecuada para el trayecto de los datos, basándose en condiciones de la red, prioridad del servicio, etc. El nivel de red agrupa pequeños fragmentos de mensajes para ser enviados juntos a través de la red.

- Nivel de Transporte. Se encarga de la recuperación y detección de errores. Garantiza también, la entrega de los mensajes del computador originados en el nivel de aplicación. Es el nivel encargado de informar a los niveles superiores del estatus de la red.

- Nivel de Sesión. Permite que dos aplicaciones residentes en computadoras diferentes establezcan, usen y terminen una conexión llamada sesión. Este nivel realiza reconocimientos de nombres y las funciones necesarias para que dos aplicaciones se comuniquen a través de la red, como en el caso de funciones de seguridad.

- Nivel de Presentación. Determina el formato a usar para el intercambio de datos en la red. Puede ser llamado el traductor de la red. Este nivel también maneja la seguridad de emisión pues provee a la red servicios como el de encriptación de datos.

- Nivel de Aplicación. Sirve como ventana para los procesos que requieren acceder a los servicios de red.

Modelo TCP/IP

La *suite* de protocolos TCP/IP permite a computadoras de todos los tamaños, marcas y sistemas operativos comunicarse entre ellas. Es un modelo en el que todos los protocolos y las aplicaciones esta disponibles públicamente sin ningún costo. Éste forma la base de lo que hoy conocemos como Internet,

Cuadro 1.2: Modelo TCP/IP

Enlace
Red
Transporte
Aplicación

una WAN que conecta a cerca de un millón de computadoras en el mundo. Los protocolos normalmente son desarrollados por capas, en el que cada capa corresponde a una parte diferente de la comunicación. TCP/IP consta de cuatro capas que se muestran en el cuadro 1.2.

- Nivel de enlace. También llamado nivel de enlace de datos, normalmente incluye el driver del dispositivo en el sistema operativo y corresponde a la tarjeta de red en la computadora. Juntos manejan todos los detalles de hardware y la conexión lógica con el cable (o cualquier medio de comunicación que se esté utilizando).
- Nivel de red. También llamada la capa de Internet, controla el movimiento de paquetes a través de la red. El ruteo de paquetes, por ejemplo, se encuentra en esta capa. IP (*Internet Protocol*), ICMP (*Internet Control Message Protocol*), y IGMP (*Internet Group Management Protocol*) también se encuentra aquí.
- Nivel de transporte. Provee el flujo de datos entre dos hosts para la capa de aplicación. En el modelo TCP/IP existen dos tipos diferentes de protocolos de transmisión: TCP (*Transmission Control Protocol*) y UDP (*User Datagrams Protocol*).
- Nivel de aplicación. Contiene los detalles de una aplicación en particular. Existen muchas aplicaciones comunes sobre TCP/IP como: FTP (*File Transfer Protocol*), telnet, SMTP (*Simple Mail Transfer Protocol*).

1.1.3. Clasificación básica de redes

Podemos clasificar a las redes de acuerdo a la extensión de éstas en las siguientes categorías:

Red de área local / LAN (*Local Area Network*)

Es una red que cubre una extensión reducida como una empresa, una universidad, una escuela, etc. No habrá por lo general dos computadoras que disten entre sí más de un kilómetro. Una configuración típica en una red de área local es tener una computadora llamada servidor de archivos en la que se almacena todo el software de control de la red así como el software que se comparte con las demás computadoras de la red. Las computadoras que no son servidores de archivos reciben el nombre de estaciones de trabajo. Estos suelen ser menos potentes y tienen software personalizado por cada usuario. La mayoría de las redes LAN están conectadas por medio de cables y tarjetas de red, una en cada equipo.

Red de área metropolitana / MAN (*Metropolitan Area Network*)

Las redes de área metropolitana cubren extensiones mayores como pueden ser una ciudad. Mediante la interconexión de redes LAN se distribuye la informática a los diferentes puntos del distrito. Bibliotecas, universidades u organismos oficiales suelen interconectarse mediante este tipo de redes.

Redes de área extensa / WAN (*Wide Area Network*)

Las redes de área extensa cubren grandes regiones geográficas como un país, un continente o incluso el mundo. Cables transoceánicos o satélites se utilizan para enlazar puntos que distan grandes distancias entre sí. Con el uso de una WAN se puede conectar desde México con Japón sin tener que pagar enormes cantidades de teléfono. La implementación de una red de área extensa es muy complicada. Se utilizan multiplexores para conectar las redes metropolitanas a redes globales utilizando técnicas que permiten que redes de diferentes características pueden comunicarse sin problema. El mejor ejemplo de una red de área extensa es Internet.

Las redes también se pueden clasificar de acuerdo al protocolo de comunicación:

Redes TCP/IP

Las aplicaciones modernas de red requieren una vía de entrada sofisticada para traer datos de una máquina a otra. Si tenemos una máquina *Unix*

con varios usuarios, cada uno de ellos deseará conectarse a un *host* remoto simultáneamente en una red, necesitamos alguna forma de permitirles ocupar las conexiones de red sin interferir con la conexión de otros. Esta vía de entrada es usualmente llamada *packet-switching*. Un paquete (*packet*) es un pequeño bulto de datos que está siendo transferida de una máquina a otra a través de la red. Una red *packet-switched* forma parte de una liga en la que los usuarios están mandando paquetes alternadamente de un lado a otro a través de esa liga. La solución que los sistemas Unix, y subsecuentemente otros no Unix han adoptado se conoce como TCP/IP. Cuando se habla de redes TCP/IP es común escuchar el término datagrama, que técnicamente tiene un significado especial pero en muchas ocasiones es utilizado indistintamente a paquete.

Redes *UUCP (Unix-to-Unix Copy)*

UUCP salió como un paquete de programas para transferir archivos sobre líneas seriales, calendarización de estas transferencias, y ejecución de programas en *sites* remotos. Ha sufrido cambios mayores desde su primera implementación en los setentas, pero aún se siguen utilizando los servicios que ofrece. Su principal aplicación son las redes WAN, basadas en ligas telefónicas *dialup*. Una de las principales desventajas de las redes *UUCP* es que operan con lotes. Mejor dicho, tienen una conexión establecida permanentemente entre los *hosts*. Un *host UUCP* debe de marcar a otro *host UUCP* sólo una vez al día, y sólo por un periodo corto de tiempo. Cuando se conecta, ésta transferirá todas las noticias, email, y archivos que han sido puestos en la cola, y después se desconectará. Este encolamiento es el que limita el número de aplicaciones que puede ser ejecutada bajo *UUCP*. En el caso del correo, el usuario puede escribir su correo y mandarlo. El mensaje estará puesto en cola en la máquina *UUCP* mientras no marque a otra *UUCP* para mandar el mensaje.

1.1.4. Topologías de red

Existen varias formas en las que podemos conectar las redes de computadoras, a esto se le llama topología de red. La topología es la configuración física de la red, la forma en que se conectan físicamente los nodos de la red. Cuando se plantea una topología, se deben de observar tres objetivos principales:

TESIS CON
FALLA DE ORIGEN

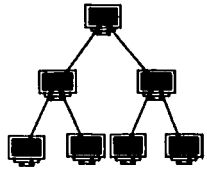


Figura 1.1: Topología Jerárquica

1. Proporcionar máxima fiabilidad que garantice la correcta recepción de todo el tráfico.
2. Encaminar tráfico entre el transmisor y el receptor a través del camino más económico y confiable.
3. Proporcionar tiempo de respuesta óptimo.

Existen cinco tipos de topología:

Jerárquica

Características:

- El software que la opera es simple y fácil.
- El servidor de mayor jerarquía es el que controla, el que maneja errores y tareas de control.

Desventajas

- Fácil que se presenten cuellos de botella.
- Saturaciones, problemas con la fiabilidad.
- Si el medio de transmisión falla deja de funcionar toda la red.

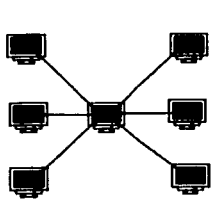


Figura 1.2: Topología de Estrella

Estrella

Características:

- Fácil de controlar, software no complicado y flujo de tráfico sencillo.
- Todo el flujo está en el nodo central que controla a todos.
- El nodo central encamina el tráfico, localiza averías y las aísla fácilmente.

Desventajas

- Hay saturaciones y problemas si se avería el nodo central.

Bus

Características:

- Frecuente en redes locales.

TESIS CON
FALLA DE ORIGEN

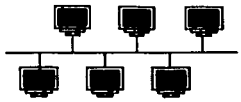


Figura 1.3: Topología de Bus

TESIS CON
FALLA DE ORIGEN

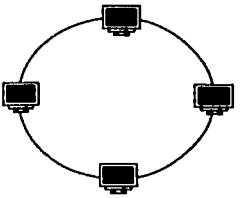


Figura 1.4: Topología de Anillo

- Fácil control de flujo en la red.
- Una estación difunde información a todas las demás.

Desventajas:

- Como hay un solo canal, si éste falla, falla toda la red.
- Casi imposible aislar averías.

Anillo

Características:

- Los datos fluyen en una sola dirección.
- Cada estación recibe los datos y la retransmite al siguiente equipo del anillo.
- Atractivo por lo raro del embotellamiento.
- Poner en marcha una topología de anillo es sencillo.
- Cada componente recibe/envía el paquete transmitido.

Desventajas:

- Como están unidos, si falla un canal entre dos nodos, falla toda la red.

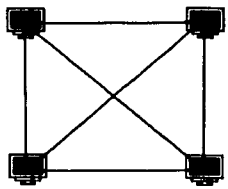


Figura 1.5: Topología de Malla

Malla

Características:

- Inmunidad a embotellamientos y averías.
- Uso de trayectorias alternativas.

TESIS CON
FALLA DE ORIGEN

Desventajas:

- Control y realización demasiado complejo pero maneja un grado de confiabilidad demasiado aceptable.

1.2. Seguridad Informática

Podemos definir a la seguridad informática como aquella en la que se encuentran protegidos nuestros datos y recursos. Existen muchas formas de violar esta seguridad, pero también existen muchas formas de protegerlas.

1.2.1. Definición

En la presente tesis, la seguridad se observa desde dos puntos de vista: funcional y estructural. Desde el punto de vista funcional se encuentran tres características importantes:

Privacidad

A ninguno de nosotros nos gustaría que nuestros archivos estuvieran publicados para todo el mundo, que gente no deseable estuviera leyendo nuestro correo o que pudiera ver nuestro estado de cuenta. La privacidad es el poder de tener mi información confidencial, y mostrar al mundo aquello que quiero compartir.

Disponibilidad

Otra de las partes de la seguridad es la disponibilidad. Supongamos que tenemos un negocio en la red el cual vende libros, no tengo ningún local en la ciudad para vender mis libros, la única parte en la que los vendo es en la red. ¿Qué pasaría con mi negocio si no estuviera disponible? Ése es un gran problema puesto que de eso vive mi negocio, perdería dinero, clientes y credibilidad. Ha esto se refiere la disponibilidad, mi información debe estar disponible en el momento en que se necesite. De igual forma la información de mis usuarios, debe estar disponible cuando ellos la necesiten.

Integridad

Otro aspecto de la seguridad es la integridad. La información que tengo almacenada siempre debe ser la misma hasta que decida modificarla. La integridad de los datos es básica ya que la razón por la que guardamos información es para tener disponible esa, y exclusivamente esa información que estamos guardando. Habría un gran problema si alguien cambiara las calificaciones, un estado de cuenta, o simplemente, si mi correo llegara alterado, pues no podría leer aquello por lo que me están escribiendo.

Éstos son los tres aspectos más importantes de la seguridad, y es por esto que los desarrolladores alrededor del mundo están trabajando tan duro para poder asegurar nuestra información.

El enfoque de la seguridad en esta tesis se orienta a todos estos puntos. Es sumamente importante la confidencialidad de la información de los usuarios. Es parte de la ética de un buen administrador el mantener privados la información de los usuarios. No importa qué tan importante sea esta información, si son sus tareas, su diario, sus estados de cuenta, el proyecto de su vida, las claves secretas para explotar bombas nucleares o sea cual fuere el tipo de información, la tarea como administrador es proteger esta información, si el

usuario quiere exponerla al público, encontraría la forma de ponerla en una página web o en algún poste para que la gente pueda verla.

Otra cuestión sumamente importante es que los usuarios puedan tener acceso a su información, o a buscar información en la red en el momento en que ellos lo decidan sin tener ningún problema. Un problema muy típico en las redes de la UNAM es que se comportan de manera intermitente. Aquí tenemos un gran problema porque muchas veces el ancho de banda está siendo utilizado por otras personas, para enviar correos *spam* o enviar un *ping* de la muerte a algún otro destino (conceptos que posteriormente se detallarán). Este problema es muy grande y debemos de analizar constantemente el tráfico de nuestra red para darnos cuenta de lo que esta pasando a través de ella.

Aquí sería importante el hacer notar que la seguridad en redes abarca desde las amenazas naturales como lo puede ser un temblor o un incendio, hasta el uso correcto por parte de los usuarios de mi sistema y la red. Podemos hacer un gran esquema de seguridad, en el que tengamos todas las herramientas que veremos en esta tesis, desde el firewall hasta sistemas de monitoreo y respuesta automática, y no nos serviría de nada si no estamos apoyados por los superiores de nuestra empresa para poder realizar buenas políticas de seguridad en nuestra red. Un *password* débil puede ser la entrada de algún intruso a nuestra red. Con esto se hace notar la importancia de la cooperación de los usuarios; la tarea de la administración no es exclusiva de los administradores, es una tarea conjunta entre los usuarios y los administradores.

La solución a este punto es la capacitación. Debemos de mostrarles a los usuarios los problemas que puede haber si no ponen passwords seguros, o si dejan su password pegado en el monitor, o si llega alguna persona y les pide permiso de usar la red y ellos le prestan su computadora. También debemos de explicarles de todos los peligros que se presentan en la Internet, y que deben de tener cuidado al transmitir archivos.

Un problema actual es el uso de programas tipo p2p, en el que los usuarios intercambian archivos de música, videos, programas, fotos, etc., por medio de la red. Éste es uno de los grandes beneficios que nos da Internet, pero debemos de tener cuidado del uso que le damos, y de tratar de hacerlo en horas en las que no obstruyamos la red por el gran tráfico que esto genera.

Como podemos ver la seguridad no es igual a decir que tu servidor es impenetrable, la seguridad abarca esto y mucho más. Existen una gran cantidad de campos en la seguridad.

Ahora, si vemos a la seguridad desde un punto de vista estructural, cons-

ta de tres etapas:

Prevención

Prevención es toda aquella acción que tomaremos para tener lista nuestra red para los ataques. Estas medidas de prevención son la instalación de los parches de seguridad y de corrección de errores, el tener nuestro sistema al día en software, el instalar herramientas para no permitir la entrada de cualquier persona a nuestra red, como lo es un *firewall*, el tener una red que no permita el que cualquier persona pueda *sniffear* nuestra red. La solución parcial a esto es una red *switcheada*. También es necesario poner a nuestros equipos herramientas que no permitan el escaneo de puertos, como *tcp-wrappers* y *portsentry*, para bloquear todas aquellas llamadas incorrectas.

Monitoreo

El monitoreo es sumamente importante, ya que el poner herramientas de prevención no es suficiente. La tarea del administrador no es estática, tiene que estar activa las 24 horas del día, y en constante actualización. El monitoreo es tanto en los servidores, como en la red. En los servidores tenemos el compromiso de revisar bitácoras todos los días y de utilizar herramientas que nos faciliten la lectura de éstas, puesto que las bitácoras de un servidor son muy grandes y en ocasiones difíciles de leer, por lo que tenemos que buscar algo que nos ayude a revisarlas. Por el lado de la red es importante el estar monitoreando las actividades, instalar herramientas que nos muestren el tráfico de la red, como *mrtg*, y herramientas que nos muestren qué se está mandando por la red, para detectar cualquier actividad rara, como los *sniffers*. Como parte del monitoreo debemos de poner pequeñas trampas a los posibles atacantes para que si caen poder dar alguna reacción a esto. Estas herramientas pueden ser los *honeypots*.

Reacción

Tener un esquema de reacción es una tarea primordial, puesto que si llegamos a captar alguna actividad extraña debemos de tener una metodología para reaccionar a ésta, que esté avalada por los superiores, puesto que puede ser desde el bloqueo de la dirección que nos está atacando, el borrar alguna cuenta del sistema por actividades extrañas, o hasta el desconectar el equipo que está siendo atacado por razones de seguridad. Un extremo en la reacción

es tener un plan en caso de que nuestro servidor se vea comprometido, el tener respaldos para poder hacer la recuperación de nuestro sistema, y un análisis forense para ver por donde entraron a nuestro sistema y no cometer el mismo error.

1.2.2. Tipos de amenazas

Existen diversos tipos de amenazas a nuestros sistemas. La seguridad de que trataremos no habla sólo de la seguridad informática. El ser un buen administrador de sistemas y de red implica todos los aspectos de la seguridad. Aquí se hablará de la seguridad física y de la seguridad lógica de nuestros sistemas, y presentaremos algunos tipos de amenazas comunes.

Eventos Físicos

A lo largo de toda nuestra vida nos hemos dado cuenta que nunca podremos predecir a la naturaleza, mucho más difícil sería el tratar de controlarla. A esto se refiere la primera parte de las amenazas. Los eventos físicos podrían ser un temblor o un incendio, en el que nuestros sistemas se vean afectados. En una gran empresa donde se espera una gran disponibilidad de los datos, se necesita realizar una planeación en la que los sistemas no estén en un mismo lugar, que si por alguna razón llegara a dejar de funcionar una estación en la que se tengan todas las bases de datos de un gran banco, exista otra en el otro extremo de la ciudad que tenga un espejo de la información. Para realizar esto debe existir redundancia de información para poder tener acceso a otro enlace con la información que necesitamos. En el caso de esta tesis no se aplicará esto, puesto que no se tienen suficientes recursos para comprar equipo de espejeo, pero se hablará de los respaldos que se tienen que hacer de la información previniendo cualquier tipo de evento. Otro gran problemas tiene que ver con el acceso a los servidores. El lugar donde tienen los servidores debe de ser un espacio con acceso fisico restringido. Es común la pérdida de información por el robo de un disco, o inclusive de una computadora completa. Otro gran problema tiene que ver con la gente que puede tener el acceso físico al servidor. La gente que tenga acceso a este lugar debe de ser gente autorizada, es preferible si sólo los administradores tienen el acceso. Se sabe de casos en los que la gente que hace la limpieza del lugar, por descuido o por desconocimiento desconectan el cable de corriente del servidor, o al estar limpiando mueven algún CPU y los discos duros se estropean

por la vibración que se generó. Es muy importante el tener cuidado físico con nuestros servidores, éste es uno de los aspectos en los que no ponemos mucha atención y pueden generar un gran número de problemas y pérdidas de tiempo y dinero.

Ataques informáticos

Ahora hablaremos de todos aquellos ataques en los que intervienen los sistemas, el software, la red, y todas aquellas partes lógicas de la computación. En el lenguaje informático la palabra *hacker* es bien conocida, pero es mal utilizada por los medios. Los *hackers* son programadores que se interesan por conocer a fondo los sistemas, que utilizan sus propios sistemas para estudiarlos y saber a detalle todas las partes que los componen. Los *hackers* son especialistas que buscan un mayor conocimiento, grandes programadores que han hecho los grandes cambios tecnológicos. Por otro lado, tenemos a los *crackers*, que son especialistas que buscan romper la seguridad de los sistemas y entrar a ellos solo por diversión o para obtener algún beneficio de esto. Ahora que sabemos quiénes son los chicos malos, podemos hablar de los problemas que nos pueden ocasionar. En esta tesis hablaremos de estos *crackers* como intrusos.

Ataques de negación de servicios

Éste es un tipo muy especial de ataque, ya que este ataque no pretende el tener el acceso y posterior control de la máquina. El ataque de negación de servicios (DoS) causa pérdidas millonarias, pues es orientado a que las grandes empresas no puedan tener acceso a su red, o que una página de web sea bloqueada y que los usuarios, redes y servicios no puedan ser utilizados. Existen varias formas de realizar estos ataques, a continuación explicaremos dos.

Consumo del ancho de banda

Éste es el DoS más común, buscará ocupar todo el ancho de banda que tiene una conexión para que no se puedan enviar o recibir más paquetes. Por ejemplo, si se tiene una conexión T1 (1.544-Mbps) y la comparamos con una carretera de 8 carriles, el intruso hará que circulen a través de ella ocho automóviles cada segundo, por lo que si alguien quiere enviar otro vehículo, la entrada estará bloqueada al tener tantos vehículos, lo que hará que nadie tenga acceso a la red.

Ping de la muerte

Es un programa que hará que muchas máquinas envíen un paquete ICMP que espera respuesta. Este paquete es para comprobar comunicaciones, lanzará una señal al sistema para saber que está funcionando, y el sistema destino le enviará una respuesta. Un ping de la muerte hará que una gran cantidad de *hosts* envíen la señal al mismo tiempo y repetidamente, hasta que el sistema destino se bloque y ya no pueda responder ninguna otra señal más que estos paquetes ICMP, por lo que no podrá responder a otros tipos de conexiones. Esto requiere de una sincronización de un gran número de *hosts* para poder enviar el ataque al mismo tiempo, y generalmente los *hosts* que atacan son máquinas que han sido violadas.

Virus

Un gran problema son los virus informáticos, éstos están afectando a un gran número de sistemas. Actualmente tenemos cerca de 60000 virus. Estos programas buscan alterar algunos archivos de sistema, borrar información, alterar, o simplemente ser reenviados a otros lugares. Los sistemas que más son afectados son los sistemas Windows, los virus son escritos con el fin de perjudicar o simplemente de diversión, y al ser retransmitidos automáticamente, afectan en poco tiempo a un gran número de sistemas.

Escaneos

El escaneo de los puertos es el primer paso que dan los intrusos para poder conocer un sistema. El escaneo de puertos consiste en utilizar herramientas que realizan conexiones a los puertos abiertos en una máquina y hacer un mapeo de los servicios que está dando esta máquina. Existen herramientas que pueden realizar este escaneo a distinto nivel. Por ejemplo, *nmap* es una herramienta que realiza conexiones y analiza la pila de respuesta del ping para poder regresar el sistema operativo de acuerdo a la forma en que responde, otra herramienta más poderosa es *snort* ya que además de dar la respuesta de los puertos abiertos y el sistema operativo, obtiene la versión del software que está dando el servicio a ese puerto y de esta manera poder buscar por vulnerabilidades conocidas de acuerdo a los *bugs* de esta distribución. Es muy importante poder detectar este tipo de problemas para poder bloquear aquellas direcciones que lo están haciendo, como se dijo anteriormente, este

es el primer paso en el manual de intrusiones.

Explotaciones remotas

Las explotaciones remotas son aquellas que se hacen por medio de la red o por medio de un servicio especial que esté prestando la máquina. Existen muchos tipos de estas explotaciones, se mostrarán algunas como ejemplos.

Ataques de fuerza bruta

Este es el tipo más lento y viejo de explotaciones: adivinar passwords a la fuerza bruta. Esto buscará algún servicio abierto que necesite autentificación por parte del usuario, como telnet y ftp. Cuando se realice la conexión intentará descubrir el *password*, y hará pruebas hasta adivinar el *password* del usuario. Este intento de adivinar el *password* irá desde utilizar fechas comunes (nacimiento, aniversario, etc.), iniciales, palabras fáciles, hasta el utilizar programas que realicen estas pruebas de manera automática, como *John the Ripper*. Este programa realizará combinaciones de letras, números y palabras de diccionario hasta adivinar el password que le permita entrar al sistema. La única solución a este problema es la revisión constante de bitácoras.

Desbordamiento de memoria

Este tipo de ataques buscará que el espacio asignado en memoria para un programa no sea suficiente, y ocupe espacios de memoria que ejecuten código especial que pueda ocasionar que el sistema ejecute este código para obtener la cuenta del administrador (*root*). Se conocen como *buffer overflow*. En mayo del 2002 hubo un gran problema de seguridad en un código que se creía seguro, *OpenSSH*. Este recibía una cadena más grande de la que podía administrar en la pila TCP/IP y ejecutaba código en un sector de memoria que era reservado para el sistema, con esto se obtenía la cuenta de *root* en una conexión.

Explotaciones locales

Son aquellas explotaciones en las que no se ve envuelta la red, que se hacen directamente en la consola, habiéndose introducido y teniendo un *shell* en el sistema. Existe una gran conexión entre las explotaciones locales y las

remotas, las remotas se vuelven locales una vez que se ha conseguido entrar al sistema, y se busca escalar en privilegios.

Ataques de shell

Uno de los grandes beneficios de los sistemas Unix es el tener un *shell* para programar al que se le pueden pasar parámetros para trabajar de cierta manera o para obtener los resultados deseados, o el poder definir ciertas variables que serán utilizadas por nuestros programas. Este es un gran beneficio, pero también un gran problema. Ha habido programas en los que al pasarle ciertos parámetros, o al definir ciertos valores para las variables, rompen la seguridad de alguna forma. Un problema muy conocido en los sistemas Sun fue que la variable interna de separación de campos (IFS), la cual está definida a espacio por default, se cambiaba por una diagonal (/) y al estar trabajando con esta variable y con un programa se generaba una confusión en el sistema que por resultado daba la cuenta de *root*.

1.2.3. Después de entrar al sistema

Ya hemos hablado de algunas formas en las que se puede entrar a un sistema, ahora veremos las herramientas que se utilizan una vez que estamos adentro del sistema.

Rootkits

Una vez lograda la intrusión, se busca tener herramientas que serán útiles en el futuro. Al tener el control de una máquina se busca que esta nueva máquina pueda ser la plataforma para futuras explotaciones a *sites* más importantes. Los *rootkits* son grupos de herramientas que me ayudarán a continuar teniendo el control de la máquina a la que se realizó la intrusión. Éste consiste de puertas traseras (*back doors*), programas falsos (*trojans*), analizadores de red (*sniffers*) y limpiadores de bitácoras de los sistemas.

Puertas traseras

Una vez que se logró el acceso a un sistema, se desea asegurar el acceso en cualquier momento, puesto que el ataque puede ser identificado o se puede cerrar el hoyo por el que entró el intruso. Una puerta trasera es un programa creado para asegurar estas entradas, se hará que programas que a cierta señal se activen y el intruso pueda volver a entrar a este sistema.

Sniffers

Estos programas son utilizados para obtener información de la red. Estos programas estarán captando toda la información que cruza su segmento de red. Básicamente capturan, interpretan y guardan todos los paquetes que se transmiten a través de la red. Esto es muy útil cuando se quieren resolver problemas de red, pero si es mal utilizado, se puede obtener información como los *login*, *passwords* e información privada de los usuarios que se está transmitiendo por la red y no utilizan protocolos seguros para mandar este tipo de información.

Troyanos

Estos son programas que substituyen a programas reales, buscan esconder información o poder obtener información a partir del uso de éstos. Tenemos dos ejemplos: El programa *ps* responde los procesos que se están ejecutando en un máquina con Unix. Estos procesos tienen asociados un usuario que nos lo permite ver y de esta forma monitorear los procesos que están ejecutando mis usuarios y qué trabajos están ocupando el tiempo de procesador y memoria de mi servidor. Si un intruso quiere esconder sus actividades, creará un troyano que no muestre los procesos que está ejecutando cuando se llame al programa *ps*. El comando *login* es el encargado de permitir la entrada a un sistema y de recibir la información para la autentificación. Se puede crear un troyano que obtenga el *login* y *password* de todos los usuarios que entren al sistema, y de esta forma tendremos una base de datos con toda esta información. La solución a esto es crear firmas electrónicas de los archivos, y monitorearlas constantemente para verificar que no han tenido ningún cambio.

Capítulo 2

Elementos de la seguridad de redes

Como se habló anteriormente, existen diversas políticas y herramientas que nos ayudarán a tener un mejor nivel de seguridad. En este capítulo se expondrán distintas ideas de esquemas de seguridad, así como las herramientas básicas que debemos de encontrar en nuestra red para poder tener un esquema de red confiable.

La seguridad se puede dividir en varias características, desde las políticas que se implanten en nuestra red, la capacitación que se le dé a los usuarios, hasta el hardware y software que se instale para forzar por un lado a los usuarios a cumplir con las políticas, hasta el tener fuera de la red a los intrusos y el contar con mecanismos de alerta y defensa contra ellos.

2.1. Estrategias de Seguridad

Es muy importante tener definido un esquema de seguridad, por lo tanto se deben de conocer los distintos tipos de esquemas que se utilizan en la actualidad. Esto es vital ya que al conocerlos se podrá tener una idea híbrida de las estrategias que se pueden emplear y la forma en que lo tenemos que hacer.

En el mundo de la seguridad los *hackers* de la seguridad tienen distintos puntos de vista de la forma en que se pueden defender contra los ataques, y los *crackers* tienen distintas herramientas para tratar de romper cada una de éstas. Al tener distintas estrategias en nuestra red, se hará una tarea más

difícil para éstos.

2.1.1. Menor Privilegio (Least Privilege)

Éste es un principio en la seguridad, no sólo en la seguridad de redes y computadoras, es un principio básico en cualquier tipo de seguridad, ya sea en casa, en el carro, en el trabajo, etc. El tener a los usuarios con la menor cantidad de privilegios es muy importante. Esto se refiere a darle al usuario sólo las herramientas que necesita, darle sólo acceso a los programas que él utilizará.

Cuando se instala un sistema *RedHat* es totalmente inseguro por default. El problema con *RedHat* es que desea poner a *Linux* al alcance de los usuarios, y evitándole problemas al escoger paquetes o al accionar ciertos servicios. Si al instalar, el administrador deja abiertas todas las puertas a los servicios, éstos podrán ser utilizados de alguna forma incorrecta. El cerrar estas puertas es una tarea sumamente difícil. *Linux RedHat 7.3* en una instalación personalizada para un servidor de correo instala cerca de 2300 comandos, de los cuales el 90% no deben de estar activos en este tipo de servidor . El revisar todos los comandos es una tarea difícil, como lo es el decidir si los usuarios deben o no tener acceso a ese comando en el servidor. El realizar esta revisión se puede llevar varios días, incluso semanas, pero el tener la seguridad de que en el servidor no se encuentran activos comandos innecesarios o algún programa que nos pueda causar un gran problema de seguridad, dará la suficiente tranquilidad para irse de vacaciones sin preocuparse porque algún usuario se vaya a poner a jugar con las herramientas a las cuales tiene acceso.

Lo anterior es por el lado de la instalación que hagamos en nuestros servidores, ¿pero qué pasa con los programas que se crean con los permisos inadecuados y qué no podemos quitar? Esto es otro gran problema de seguridad en el diseño de los programas que se utilizan. Un ejemplo claro es el *lpd*, *lpd* es el demonio de impresión. Cuando se desea imprimir algo, se le manda la señal a *lpd*, quien es el encargado de mandar la impresión a la impresora. El problema con *lpd* es que debe de estar corriendo con *setuid* a root, esto quiere decir que el encargado de hacer la impresión será root, y no el usuario que desea imprimir. Todos los programas que tienen *setuid* hacia root son un problema de seguridad, puesto que si llegan a tener un pequeño error en la programación podrán realizar corridas de programas como root. Algunos desarrolladores, como los de *OpenBSD* se han preocupado por esto, y desde

la versión 3.1 de *OpenBSD lpd* ya no corre como *setuid*. Este problema se ha resuelto en este sistema, pero todavía existen otros, por ejemplo *sendmail*. *sendmail* es el encargado de enviar y recibir correos, pero es un programa enorme, con una gran cantidad de líneas de código que corren como *setuid* a root, y al ser tan grande es más susceptible a errores de programación.

2.1.2. Defensa a fondo (Defense in Depth)

La seguridad debe de ser redundante. Se deben de instalar diversas herramientas de seguridad que complementen a otras, buscando que si en algún momento rompen nuestra barrera de seguridad, exista otra que pueda ayudar a alejar a los intrusos. El tener diversas herramientas de seguridad busca hacer la tarea de intrusión más difícil y costosa para el intruso, y de esta manera alejarlo. Esto se puede realizar instalando diversos mecanismos que hagan la redundancia de la que estamos hablando: seguridad de redes (como un *firewall*), seguridad en *host* (particular en cada uno de los *hosts*), y seguridad humana (educación de los usuarios, administración cuidadosa de los sistemas, etc.). Todos estos mecanismos son importantes y efectivos, pero no se debe de poner una confianza absoluta a ninguno de ellos.

Un error común en la instalación de las herramientas de seguridad es bloquear ciertos tipos de paquetes en el *firewall*, pero dejar abierto el servicio en el *host*. Lo que debemos de hacer notar aquí es el hecho de que si no vamos a proveer el servicio de correo en el *host* específico, y lo bloqueamos en el *firewall*, ¿Para qué dejar funcionando el servicio en el *host*? Esto puede suceder, por ejemplo, en el servidor de archivos. A lo que nos exponemos con esto es que exista un error de programación en *sendmail*, y al tenerlo bloqueado en el *firewall* no puedan existir ataques externos, pero puede existir un intruso interno que trate de quebrar la seguridad por medio de ese paquete. No tenemos porqué tener corriendo el servidor de correos en el servidor de archivos, al quitar totalmente *sendmail* de nuestro servidor de archivos, nos dejaremos de preocupar por los posibles errores que tenga *sendmail* y de esta manera tendremos la seguridad a nivel del *firewall* y a nivel del *host*.

2.1.3. Punto de ahogo (Choke Point)

Esta estrategia se basa en tener un cuello de botella, en la cual podamos analizar todo lo que entra y sale de nuestra red. Debemos de poder monitorear y bloquear aquello que no deseamos que pase a nuestra red. Tratemos de ver

un esquema de red en el que todo el tráfico que viene de *Internet* debe de pasar a través de una máquina, la cual podamos estar monitoreando. Ésta es la función de un *firewall*, todo el tráfico que viene de Internet debe de pasar a través de él, y de esta manera se pueden bloquear todos los posibles ataques que vienen de *Internet*. Esto es sumamente efectivo, pero debemos de tener cuidado de no tener otras entradas por las que se pueda pasar fácilmente y con esto se le dé la vuelta a nuestro punto de ahogo, y debemos de tener listos esquemas de reacción para cuando nuestro *firewall* detecte algún intruso entrando en nuestra red.

Puede haber una variante de este esquema, en la que seccionemos nuestra red en redes más pequeñas, dependiendo de lo que deseamos filtrar, y poner en cada una de éstas un punto de ahogo, para poder filtrar de distintas formas el tráfico que está tratando de llegar a los *hosts* de esta pequeña avenida.

2.1.4. Eslabón más débil (Weakest Link)

Un principio básico en la seguridad es que una cadena es tan fuerte como su eslabón más débil, y un muro es tan fuerte como su punto más débil. Los atacantes buscarán siempre el punto más débil en nuestra red, para tratar de romper la seguridad de ella. Es necesario que como administradores y encargados de la seguridad sepamos cuáles son nuestros puntos débiles para de esta manera tratar de eliminarlos y concentrarnos en los puntos que sabemos que pueden ser más atractivos para los atacantes. En este caso podemos incluso ponerles trampas, como los son los *honeypots*. Los *honeypots* son programas que simularán tener vulnerabilidades que son conocidas y han sido parchadas, para de esta manera llamar a los intrusos a estas máquinas y poder tener conocimiento de que de algún lado están buscando esa vulnerabilidad en nuestra red, de esta manera se podrá tener conocimiento del ataque, e incluso tener un esquema de reacción ante esto.

2.1.5. Postura de falla segura (Fail-safe Stance)

Éste es otro principio básico de la seguridad. El pensar en que exista una falla, y que cuando exista la falla se pase automáticamente a un modo seguro. Esto suena un poco difícil de entender, pero lo explicaremos con un ejemplo. Supongamos que tenemos un servidor de correos que sólo permite conexio-

nes seguras[1] y repentinamente deja de funcionar el servidor de conexiones seguras (como *ssh*). Lo que esperamos de este servidor es que deje de permitir conexiones, y que desconecte a todas los usuarios que están dentro de nuestro servidor, puesto que si deja a los usuarios que ya están conectados, éstos podrían estar realizando la conexión de una manera insegura, o si permite conexiones de manera insegura, nuestro esquema de seguridad no habrá funcionado adecuadamente, y estas conexiones inseguras podrían ser de un *cracker* que está entrando en nuestro sistema.

La aplicación más grande de este principio viene con la postura de la seguridad en tu red. La postura, es esencialmente, la postura de tu sitio frente a la seguridad. Tu postura puede ser restrictiva o permisiva. ¿Estás del lado de la seguridad (incluso hay quien la llama paranoia) o de la libertad?

Aquí están dos posturas fundamentales respecto a las decisiones y políticas de la seguridad:

La postura de negación por default En esta postura se define qué estará permitido en nuestra red. Definiremos los protocolos, tipos de conexiones, direcciones a las que se está permitido, puertos a los que puedo realizar conexiones, sitios desde los que me puedo conectar, etc., y negaremos por default todo aquello a lo que no estemos dando permiso explícito.

La postura de permiso por default Esta postura permitirá todo tipo de conexiones, y bloqueará sólo aquellas que nosotros explícitamente estemos cerrando. Ésta es mucho más sencilla, pero desde el punto de vista de la seguridad es totalmente inadecuada. El mundo de *Internet* crece rápidamente, mucho más rápido de lo que nos podemos imaginar. Día a día se escriben nuevos programas, se descubren nuevos *bugs*[2], se encuentran nuevas vulnerabilidades, etc., y por más que un administrador trate de estar actualizado, siempre existe un tiempo en el que no sabemos de estos problemas, y esto puede ser aprovechado por los intrusos para tratar de entrar a nuestros sistemas con este nuevo programa, o engañar a nuestros usuarios y abrir una puerta hacia nuestra red.

[1]Las conexiones seguras son aquellas en las que si algún intruso llega a capturar el tráfico, éste no podrá ser entendido ya que viene cifrado por algún algoritmo como ssh.

[2]Errores de programación

2.1.6. Participación universal

Éste es el punto medular de la seguridad informática y de redes: la participación de todos los usuarios. De nada nos sirve poner un gran esquema de seguridad, con sistemas detectores de intrusos, *firewalls* que incluso nos pudieron costar varios miles de dolares, gente dedicada las 24 horas del día a estar supervisando y protegiendo nuestra información, si no contamos con la cooperación de todos los usuarios. ¿De qué nos serviría poner un esquema de cambios de *password* si cada vez que mi usuario lo cambia lo publica en Internet?

Esta participicación puede ser de dos formas: voluntaria u obligatoria. La voluntaria es aquella que por medio de la capacitación y la información que demos a nuestros usuarios, se den cuenta de lo importante que es para todos, usuarios y red, la seguridad. Ésta se tiene que dar por medio del convencimiento de los administradores hacia los usuarios. La obligatoria tiene que venir de una autoridad con el suficiente poder para poder exigirles a los usuarios que se comporten de cierta forma. También se puede hacer una combinación de ambas, lo cual nos daría un mayor respaldo, pero siempre es mejor la participación voluntaria, en la que los usuarios se den cuenta de los problemas que puede acarrear el no cumplir con las normas básicas de seguridad.

El tener de nuestro lado a todos los usuarios es vital, puesto que un usuario que no esté de acuerdo con nosotros puede ser un gran problema. El usuario rebelde puede hacer cosas que estén dentro de las reglas que puedan afectar a nuestros sistemas y a nuestra red, y la única forma de evitar esto es haciendo que los usuarios nos ayuden con la seguridad, que nos informen de cualquier problema extraño que tengan en su máquina y pueda estar relacionado con la seguridad, así como el que nos mantengan informados de cualquier cosa extraña que puedan detectar con la velocidad de la red. Un buen ejemplo sería que alguien estuviera imprimiendo el archivo de *passwords*, lo que esperaríamos es que nuestros usuarios nos informarán de esto. ¿Para qué querría alguien el archivo de *passwords* de nuestro servidor de correos?

2.1.7. Simplicidad

La simplicidad es una estrategia de seguridad por dos razones. La primera es que manteniendo las cosas simples las hace más fáciles de entender; si no entiendes algo, no puedes saber si es realmente seguro. La segunda es que

la complejidad provee de rincones en donde se pueden esconder cualquier cantidad de cosas.

Los programas complejos contienen más errores, muchos de los cuales pueden ser problemas de seguridad, siempre se ha sabido que el tener una tela sin hoyos ni parches es mejor que el tener una tela llena de parches, como lo es un sistema al que le vamos poniendo los parches del software al sistema original.

2.1.8. Seguridad a través de obscuridad (Security Through Obscurity)

La seguridad a través de obscuridad se basa en el principio de proteger las cosas ocultándolas. Esto se aplica a la seguridad de redes y sistemas, pues se puede mantener totalmente en secreto el sistema de seguridad, inclusive si éste no existe. En términos computacionales, tenemos los siguientes ejemplos de este tipo de seguridad:

- Poner una computadora con conexión a *Internet* y pensar que nadie tratará de romper la seguridad pues nadie sabe que está ahí.
- Desarrollar un nuevo algoritmo de encriptación de datos y no mostrárselo a nadie más.
- Correr un servicio en un puerto diferente al que se utiliza normalmente.
- Configurar tu *firewall* para que los usuarios externos no vean la misma información que pueden ver tus usuarios internos.

Los *hackers* dicen que la seguridad a través de la obscuridad es mala. Podemos decir que es un medio válido, siempre y cuando venga acompañado de otras herramientas que le ayuden a tener un buen sistema de seguridad. La seguridad a través de la obscuridad es mala cuando:

- Es la única seguridad existente.
- No existe nada de obscuridad alrededor de ésta.
- Le da a los usuarios una confianza irracional.

Con esto no se quiere decir que el tener nuestro sistema de seguridad privado es malo. Entre menos información tengan los intrusos es mejor, al no dar esta información públicamente haremos la tarea de intrusión más difícil, que es lo que se está tratando de hacer. Se puede dar una conferencia sobre la instalación del *firewall* y dar buenos tips de seguridad sin tener que dar a detalle nuestro esquema de seguridad y las reglas que implantamos. No se desea que los atacantes sepan:

- Exactamente qué tipo de equipo y herramientas estamos utilizando en nuestro *firewall.*
- Qué protocolos estamos permitiendo bajo qué condiciones.
- Nombres válidos de los usuarios y *hosts* internos.
- Qué tipo de detectores de intrusos estamos utilizando.

2.2. Herramientas de seguridad a nivel de *host*

Ahora que ya tenemos una mejor idea de los tipos de seguridad que podemos tener en nuestra red, hablaremos de las herramientas que nos podrán ayudar a tener una conjunción de todos estos esquemas de seguridad. Hablaremos de la seguridad que debe de haber en cada *host* así como de las herramientas que se deben de instalar en la red para poder monitorearla.

Inciaremos con las herramientas que podemos instalar en nuestros *hosts.*

2.2.1. Tipos de servidores

Existen muchos tipos de servidores en el ambiente de redes, en realidad puede existir un servidor para cada servicio de *Internet.* Se pueden encontrar servidores de correo, de web, de información de claves, de archivos, de direcciones dinámicas, de cálculo, etc.

Lo principal en los servidores son las configuraciones que se les dé, que se tengan las herramienta de seguridad necesarias para el servicio, tener únicamente abiertos los puertos que son indispensables para ese servicio, tener los demonios necesarios y no otros que estén ocupando recursos del sistema o que puedan ser puertas de entrada para posibles ataques. Por ejemplo, en los servidores Unix se puede tener un servicio llamado *NFS* (*Network File*

System) que es el servicio encargado de compartir archivos en una red Unix. Para tener funcionando este servicio se tiene que levantar un demonio conocido como *mountd*, que es parte de los paquetes de *RPC* (*remote procedure calls*). Otro de los servicios que necesita tener habilitado nuestro servidor de archivos es *portmap*, que es el encargado de administrar las llamadas de los servicios *RPC*. Como se vio en este caso, para poder tener habilitado el servicio de servidor de archivos se necesitan tener varios demonios arriba, y muchos de ellos pueden ser inseguros como lo es por naturaleza el *nfsd* y el *mountd*. Es por eso que es sumamente importante tener funcionando unicamente los servcios indispensables para proveer el servicio para el que está diseñado nuestro servidor. El tener demonios o servicios arriba innecesarios son entradas a errores del sistema, entre menor sea el número de estas entradas, menor será la posibilidad de que alguno de nuestros programas tenga algún tipo de error que dé la entrada a nuestro servidor.

Tenemos que ser cuidadosos con varias partes de la instalación y configuración de nuestros servidores, a continuación hablaremos un poco de estos procedimientos y las caracterísitcas importantes en las que tenemos que fijar nuestra atención, así como las instalaciones especiales de nuestros servidores y las herramientas que utilizamos en éstos.

2.2.2. Instalación

Siempre es importante iniciar bien con un trabajo, y el trabajo de la seguridad de los servidores se debe empezar por una buena instalación. La instalación es un proceso muy importante, pues se deben de tomar en cuenta muchas cosas para tener una buena base. Es importante que los servicios que se instalen sean los necesarios para nuestro servidor.

Comenzando por la planeación de las particiones nos podemos dar cuenta en que si el servidor es de archivos, necesitará un mayor espacio para los archivos de los usuarios. Si el servidor es de correos, se necesitará un mayor espacio en la partición en la que se guardan los correos cuando llegan al servidor, y también en la partición en la que los usuarios guardarán sus correos cuando lo necesiten. El tener bien particionado un disco duro es sumamente importante por la integridad de los sistemas de archivos. Un sistema de archivos debe de procurar estar siempre abajo del 90% de su capacidad, esto para evitar cualquier tipo de bloqueos. Cuando un sistema de archivos está lleno deja de recibir información, por ejemplo si la partición en donde se guardan los correos de entrada se llena, el servidor de correos comienza a en-

viar los correos de regreso al recibirlos con el mensaje de sistema de archivos lleno, y recordando que la disponibilidad es uno de los puntos importantes de la seguridad, en este punto estamos teniendo un gran problema.

Otro punto importante es la instalación de los paquetes, el tener un servidor con paquetes innecesarios instalados es otro gran problema de seguridad. La tendencia de los sistemas operativos libres es que los usuarios comunes puedan utilizar los sistemas operativos en sus máquinas de escritorio, y para esto necesitan hacer una instalación más fácil y amigable. Actualmente RedHat y Mandrake tienen las interfaces de instalación de sistemas operativos libres más amigables, pero esto no quiere decir que sean las mejores. RedHat tiene en su menú de instalación la opción de "estación de trabajo", la de "servidor" y la "personalizada". Si vamos a instalar un servidor lo mas lógico es que se tomara la opción de "servidor", pero ésta no es la mejor opción. Decir servidor es algo muy vago, podemos decir que es un servidor de archivos, un servidor de correos, un servidor de web, un servidor de direcciones dinámicas, un servidor *proxy*, etc. Al no poder especificar qué tipo de servidor es el que instalaremos, esta opción instalará todas las anteriores, y esto es justo lo que tratamos de evitar al realizar nuestra instalación. La selección de los paquetes debe de ser detallada y orientada totalmente al servicio que se brindará.

Hablando de los errores importantes en la instalación encontramos otro error muy común: la instalación de los paquetes para el ambiente gráfico. *X* (el nombre con el que se le conoce al ambiente gráfico de los Unix) es un programa muy grande, en realidad es enorme, y al ser tan grande está lleno de errores. *X* tiene errores desde su planeación, puesto que para tener *X* funcionando se abre un puerto especial para éste, el puerto 6000, al cual se le puede hacer un *hackeo* y robarse la sesión de *X*, es por esto que es sumamente importante que en los servidores no se tenga levantado el ambiente gráfico, y de ser posible no tener instalada toda la paquetería de *X*, pues recordemos que el tener más líneas de código instalada, hará a nuestro sistema más vulnerable a posibles errores de programación.

2.2.3. Actualizaciones

El tener el sistema actualizado es vital para la estabilidad y seguridad de nuestro sistema. Todos los días se descubren errores en los programas y el tener el sistema actualizado es la mayor herramienta de seguridad. La cantidad de sitios *hackeados* de acuerdo al tiempo de aparición de los parches se muestra en gráfica 2.1.

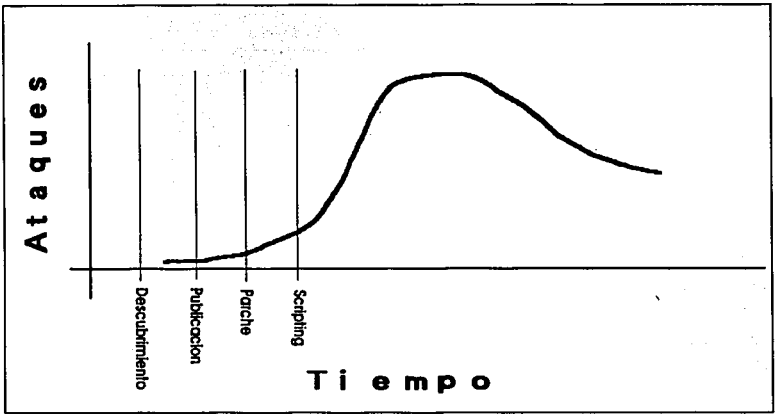


Figura 2.1: Sistemas *hackeados* respecto al tiempo

Cuando se encuentra un *bug* en cualquier tipo de programa y se reporta a alguna de las listas de *hackers* existe un pequeño incremento en el número de *hosts* atacados, por las pruebas de los *hackers* que están suscritos a estas listas y de algunos *crackers* que al enterarse de este error en la programación que fue encontrado por alguna persona en el mundo, tratan de tomar ventaja de éste. Después viene el momento en el que la empresa o los programadores encargados de ese programa liberan la solución y el parche del error, y en este momento es en cuanto empiezan a salir los *scripts* de automatización de la explotación del *bug*, es por eso que es tan importante tener actualizado nuestro sistema, pues en el momento en que se hace público el error de programación y al tener el código que lo corrige, los *crackers* pueden encontrar fácilmente el error, hacer un *script*, ponerlo en la red y permitir que varias personas jueguen con éste. Cómo lo hablamos en un inicio, los *crackers* más peligrosos son aquellos que lo hacen por jugar, que encuentran algún *script* en la red y lanzan ataques a nuestras redes, y al no saber lo que están haciendo pueden ocasionarnos un gran problema.

Cada empresa o desarrolladores del sistema operativo deben de tener su propio sitio de parches, por ejemplo, los parches para RedHat se pueden

TESIS CON
FALLA DE ORIGEN

encontrar en:

http://www.redhat.com/apps/support/errata/

y los de OpenBSD los puedes encontrar en:

http://www.openbsd.org/errata.html

Junto con los parches encontraremos la página que nos indica las instrucciones de instalación. También es importante bajar los parches de un lugar confiable, no podemos experimentar con bajar parches de algún lugar que nos promete ser más rápido, pues como los parches se tienen que instalar como administrador, si instalamos algún parche falso, puede que éste sea algún tipo de programa malicioso, como un *back door*.

Las empresas están tratando de hacer más fácil la administración de los Unix, pues éste era uno de los principales problemas que tenian los usuarios. Ahora algunas empresas como RedHat tienen actualizadores automáticos, por lo que se puede utilizar *up2date* para hacer las actualizaciones de una manera más sencilla.

2.2.4. Configuración

Ahora que ya se tienen los programas actualizados, que se ha realizado una instalación a conciencia, el siguiente paso es verificar que sólo estén funcionando los servicios que se necesitan. Cada versión de Unix tendrá sus propios archivos de configuración para los servicios que deben o no estar funcionando en un servidor. Tenemos que estar concientes de cada servicio que levantamos en nuestro servidor, para tener la seguridad de que nuestros sistemas están funcionando correctamente y que no tenemos puertas abiertas innecesariamente.

El problema con las configuraciones también es independiente en cada servicio. Por ejemplo, si tenemos un servidor de *ftp* debemos de verificar que no acepte conexiones anónimas. O si las acepta, tenemos que comprobar que sólo puede navegar en los directorios que ponemos públicos. Hubo un error muy famoso en el *wu-ftpd* en el que permitía navegar sobre todo el árbol de directorios, el problema era que por configuración no permitía darle rutas desde la raíz (/), por lo que no se podía mover a directorios como /etc. Pero sí permitía moverse al directorio padre, por lo que se permitía dar una ruta como ésta:

../../../etc

y esto permitía moverse a cualquier directorio en el servidor. El siguiente paso era traer archivos como el *passwd* o el *shadow* e intentar romper alguno

de los passwords por la fuerza.

2.2.5. Herramientas

Aquí se hablará de las herramientas básicas de seguridad para el servidor.

Tripwire

Ahora se comenzarán a tratar algunas herramientas básicas (es decir, no únicamente) que debe de tener cualquier servidor.

Tripwire es una herramienta que nos permite revisar los cambios en nuestros sistemas. En un principio *tripwire* genera una base de datos de los archivos de configuración y programas del sistema, tomando en cuenta su nombre, fecha de modificación, permisos, tamaño, etc. Con estos datos genera firmas digitales de cada archivo, lo que garantiza que los datos que están en la base de datos no puedan ser modificados. Cuando se le pide a *tripwire* que verifique los cambios en el sistema hace un recorrido por todos los archivos que tiene en la base de datos comparando las firmas digitales, si alguna de éstas ha cambiado desde la última actualización, *tripwire* genera un reporte en el que nos dice cuáles han sido los cambios en el sistema o en los programas de los que tenía firmas digitales.

Esto evita el problema de los troyanos. Una vez que un intruso entró al sistema, intentará dejar programas en los que el administrador no pueda detectar que él está conectado, con esto esconderá su actividad y podrá seguir utilizando el servidor para sus propósitos.

TCP-Wrappers

TCP-Wrappers es una herramienta que nos ayudará a controlar el acceso a ciertos servicios de red. *TCP-Wrappers* puede controlar el acceso a todos los servicios de red que son controlados por *xinetd.* Por medio de los archivos de configuración */etc/hosts.allow* y */etc/hosts.deny* se pueden bloquear los servicios para una red, o un segmento de red.

Por ejemplo, necesitamos que los usuarios que están dentro de nuestra red puedan tener acceso al correo electrónico por medio de lectores comúnes que se pueden ejecutar bajo algún otro sistema operativo. Para esto necesitamos activar los servicios de *pop3* e *imap*, que son levantados desde el *xinetd.* Al tener arriba estos servicios, cualquier usuario se podría conectar desde fuera de nuestra red, e incluso podría ser un intruso que está intentando

entrar a nuestro sistema. Justo para esto sirve *TCP-Wrappers*, porque con él podemos decir que estos servicios estarán funcionando, pero que sólo gente desde nuestra red, o desde algunas direcciones especiales que nosotros demos permiso se pueden conectar.

El problema con *TCP-Wrappers* es que sólo funciona para los servicios que son controlados por *xinetd*, y que trabaja en un nivel de red muy alto, por lo que podemos bloquear servicios, pero no conexiones que estén intentando romper la seguridad de nuestro sistema por medio de algún servicio que no esté en ese nivel de red.

Portsentry y logsentry

Portsentry es una herramienta que nos ayuda a detectar escaneos en los puertos o conexiones. *Logsentry* es una herramienta que nos ayudará a organizar las bitácoras para facilitar la lectura de éstas. Gran parte del tiempo del administrador es la revisión de las bitácoras, que es una tarea difícil y que a cualquiera puede hartar, pero en seguridad es una de las tareas más importantes. Una de las grandes características de los sistemas Unix es la posibilidad de poder configurar lo que se quiere en bitácoras y lo que no se necesita, y las conexiones que se hacen a nuestros servidores siempre debe de estar monitoreado, para poder darnos cuenta de cualquier actividad extraña.

Como lo dijimos anteriormente, la revisión de las bitácoras es una tarea demasiado pesada, puesto que se generan un gran número de líneas en nuestras bitácoras. Por ejemplo guardar los encabezados de un paquete para verificar una consulta posterior, si tenemos un servidor de correos en el que nuestros usuarios se pueden conectar por medio del puerto *pop* y nuestros usuarios les gusta checar su correo cada 10 minutos (que los administradores sabrán que esto es un periodo muy largo, existen usuarios que lo configuran para verificar cada minuto) esto generaría un par de líneas cada diez minutos por usuario; si pensamos que nuestro servidor atiende a cerca de 300 usuarios, generaría cerca de 600 líneas cada diez minutos, lo que nos da un total de 3600 líneas por hora!!!

Y esto es sólo para el servicio de correo, falta verificar las conexiones a los correos de entrada, de salida, las bitácoras propias del sistema, las del web, etc. Lo que nos genera archivos enormes. Como podrá darse cuenta, encontrar un intento de ruptura de contraseña en estos archivos, es como encontrar una aguja en un pajar.

Para esto es lo que nos sirven estos dos programas. *Portsentry* buscará por

actividades comunmente utilizadas por los intrusos para romper la seguridad del sistema, alertará al administrador en caso de detectar un escaneo de puertos, o la llamada a un puerto "trampa" en el que estará esperando alguna llamada para mandar el reporte.

Logsentry nos ayudará a la lectura de las bitácoras, tirará a la basura todo aquello que podemos considerar como bitácoras "normales" del sistema, y nos mandará un reporte amigable de las actividades extrañas en nuestros sistemas. Ambas herramientas son configurables y adaptables a varios Unix, por lo que podremos fácilmente adaptarlo a nuestros servidores.

2.2.6. Servidor de correos

El servidor de correos es uno de los puntos más llamativos en una red. A cualquier lado que vamos damos la dirección de email, es por esto que nuestro servidor será uno de los principales objetivos, por lo que debemos de tener especial cuidado.

Además de tener el cuidado del servidor, se debe de tener cuidado con uno de los grandes problemas actuales de la seguridad: los virus. Los virus son un gran problema sobre todo del sistema opertivo Windows. Windows es un sistema operativo con un pésimo diseño de seguridad, por lo que un correo con virus puede hacer que nuestro sistema se vuelva totalmente inestable, que perdamos información, o que nuestro sistema se convierta en un sistema que esté monitoreando nuestra red en busca de errores, es por eso que debemos de tener tanto cuidado con los virus. Existen virus que pueden radicar en nuestro sistema sin que nosotros nos demos cuenta, y estar captando la información que yo envié a tráves de mi teclado o a algún lugar por medio del correo o cualquier otro protocolo. El correo es una de las principales entradas de virus a nuestros sistemas, es por esto que debemos de poner un bloqueo a los virus desde nuestro servidor de correos. En esta tesis hablaremos de la instalación de un analizador de correos llamado *mailscanner* y un antivirus llamado *f-prot*. Lo que haremos con esto es poner un filtro de entrada a los correos, antes de entrar a mi red y de ponerlo disponible para los usuarios, los revisaremos contra cualquier problema de virus.

Mailscanner

Mailscanner es un sistema de seguriad integral para e-mail diseñado para ser utilizado en servidores de correo. *Mailscanner* puede proteger a los usuarios contra virus, y detectar ataques contra paquetes de correo (como *Eudora,*

Outlook, Outlook Express). *Mailscanner* también puede detectar el correo comercial no solicitado (conocido como *spam*) y tomar acciones diferentes en cada uno de los casos. Estas acciones pueden ser desde bloquear totalmente el correo y tirarlo a la basura, enviar un aviso al destinatario, otro al administrador y otro a la persona que lo envía avisándole del problema del correo, tratar de desinfectar el correo, crear estadísticas, etc.

No sólo busca por virus conocidos, también puede proteger contra virus que no han sido discubiertos en un e-mail, esto lo hace por los nombres de los archivos que tratan de ocultar su extensión (por ejemplo ".txt.vbs").

Los archivos adjuntos a un e-mail pueden ser desinfectados automaticamente y ser enviados a destinatario original.

Mailscanner es sólo la máquina que realiza la búsqueda por virus en los correos, necesita de un antivirus que le ayude a encontrar los virus y las posibles vacunas. *Mailscanner* acepta una gran cantidad de antivirus, tanto comerciales como libres, por lo que podemos hablar de que *Mailscanner* es flexible y nos da la posiblidad de elegir el antivirus que utilizará.

F-prot

F-prot es un antivirus creado por la empresa *F-secure*, por el momento este software es libre y puede ser bajado de la página de Internet de la compañía, así como las actualizaciones pueden ser obtenidas desde el sitio *ftp*. Este antivirus puede ser utilizado en cualquier compañía, y ha resultado de gran ayuda pues tiene cerca de 60000 definiciones de virus que puede detectar. Más adelante se mostrara un *script* que se puede utilizar para bajar actualizaciones cada cierto tiempo, lo que nos dará la seguridad que si en algún momento se descrubre un nuevo virus, en poco tiempo quedará actualizado nuestro antivirus automáticamente.

2.3. Herramientas de Seguridad a nivel de red

Ahora que hemos hablado de la seguridad en los *hosts*, hablaremos de la seguridad que debe de haber en nuestra red, cómo podemos controlarla y monitorearla, para de esta manera tener seguridad en todos nuestros equipos.

La seguridad de la red es una tarea sumamente difícil ya que podemos tener una gran cantidad de equipos en nuestra red, y no podemos estar revisando uno a uno todos los días, por lo que se debe de tener un esquema de red seguro y las herramientas necesarias para monitorearla. Las herramientas

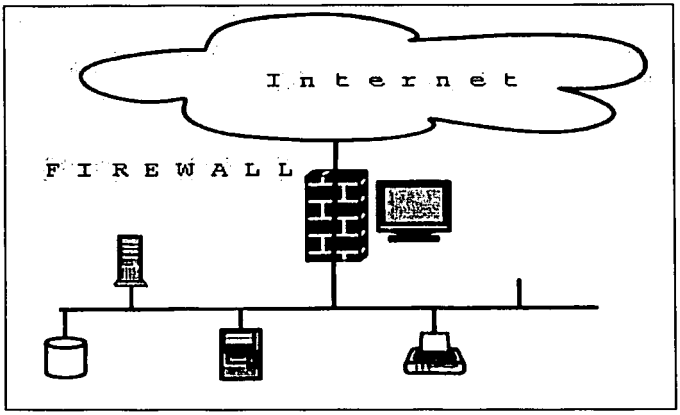


Figura 2.2: Esquema de Firewall

de red son un arma de doble filo, pues estas mismas herramientas son las que utilizan los *hackers* para monitorear nuestra red y buscar las vulnerabilidades, es por esto que nosotros debemos de saber de esas vulnerabilidades antes que los intrusos, para poder prevenir los ataques.

2.3.1. Firewalls

Los *firewalls* pertenecen a una tecnología de moda, actualmente todas las empresas de seguridad recomiendan tener uno, pero también hay una falsa creencia de que los *firewalls* son las panacea de la seguridad, creyendo que al tener un *firewall* las computadoras y su información estarán totalmente seguras.

Podemos definir al *firewall* como "un componente o conjunto de componentes que restringe el acceso entre una red protegida e *Internet*, o entre diferentes grupos de redes". [3] En la siguiente gráfica mostraremos cuál es la configuración básica de un *firewall* y cómo se crea una zona protegina por él.

Ya que se ha hablado de la definición del *firewall* y de cómo proteger

[3]Tomada del libro "Building Internet Firewalls" Pag. 102

TESIS CON
FALLA DE ORIGEN

nuestra red con él, entraremos a más detalle en las configuraciones que se pueden dar a este tipo de redes, para esto se explicarán las tecnologías más utilizadas en la seguridad.

Filtrado de paquetes

Los sistemas de filtrado de paquetes direccionan paquetes entre *hosts* internos y externos, pero lo hacen selectivamente. Éstos permiten o niegan ciertos tipos de paquetes de tal forma que cumplan con las políticas de seguridad del sitio. El tipo de direccionamiento utilizado en un *firewall* de filtrado de paquetes es conocido como *screening router*.

Para entender el filtrado de paquete tenemos que hablar de los encabezados de un paquete *TCP/IP*. Cada paquete que llega a través de la red contiene cierta información, la información más importante de los paquetes es:

- Dirección IP de origen
- Dirección IP destino
- Protocolo (TCP, UDP o ICMP)
- Puerto de origen TCP o UDP
- Puerto de destino TCP o UDP
- Tipo de mensaje ICMP
- Tamaño del paquete

El *router* también puede permitir la salida de paquetes que son solicitados por alguna otra llamada futura; esto permite realizar filtrados de paquetes en una información más detallada (como el nombre de la página web que alguien está preguntando) y si los paquetes tienen un formato adecuado para el puerto al que se dirigen. El *router* también puede asegurarse que el paquete es válido (de hecho verifica que sea del tamaño que dice ser y que sea un tamaño permitido), lo que ayuda a atrapar algunos ataques de negación de servicios basados en paquetes malformados.

Además, el *router* conoce cosas del paquete que no son reflejadas por el mismo paquete, como:

- La interfaz a la que llega el paquete

- La interfaz en la que el paquete deja el *firewall*

Finalmente, un ruteador puede guardar la pista de los paquetes y saber algunos hechos históricos, como:

- Cuándo este paquete aparezca como respuesta de otro paquete (el orgien fue el destino de un paquete reciente y el destino es el origen del otro paquete)
- Cuántos paquetes han sido enviados o han llegado desde el mismo *host.*
- Alguno de estos paquetes es idéntico a algun otro paquete enviado.
- Si el paquete es parte de otro más grande que ha sido partido en pequeñas partes (fragmentación)

Para entender la forma en que trabaja el filtrado de paquetes explicaremos las diferencias entre un ruteador tradicional y un *screening router.*

Un ruteador tradicional unicamente trabaja con la dirección de destino de cada paquete y señala la mejor dirección que él conoce para llegar al destino. La decisión de cómo manipular el paquete se basa únicamente en la dirección de destino. Aquí existen dos posiblidades, o el ruteador sabe cómo mandar el paquete hacia su destino, o el ruteador no sabe hacia dónde mandarlo, y se olvida del paquete y manda un mensaje "destination unreachable", hacia el origen del paquete.

Un *screening router* mira más profundamente en el paquete. Además de determinar si puede o no dirigir el paquete hacia su destino, un *screening router* también determina si lo "permite". "Permitirlo" o "no permitirlo" está determinado por las políticas de seguridad del sitio, para lo cual está configurado este *firewall.*

El filtrado de paquetes debe también estar compuesto por dispositivos solo al "deber" o "no deber" y no tener la posibilidad de hacer un ruteo. Los dispositivos que saben realizar esto son los puentes de filtrado de paquetes. Son como ruteadores de filtrado de paquetes, ya que son dispositivos dedicados totalmente a la seguridad, y le agrega características importantes al ruteo.

Una vez que el *firewall* tiene esta información, éste puede realizar alguna o varias de las siguientes acciones:

- Enviar el paquete hacia el destinatario

- Tirar el paquete -sólo olvidarlo, sin ninguna notificación al remitente
- Rechazar el paquete -rechazar el enviarlo, y enviar un error al remitente
- Guardar en bitácoras información acerca del paquete
- Enviar alguna alarma para notificar inmediatamente a alguien

Proxy

En general un *proxy* es algo o alguien que hace algo en favor de alguien. Los servicios *proxy* son una aplicación especializada o servidores de programas que toman las peticiones del usuario de los servicios de *Internet* y los reenvían a los servidores actuales. Los *proxies* realizan la función de ruteadores, toman la información y la reemplazan. Por esta razón los *proxies* son conocidos como *gateway* a nivel de aplicación.

Inicialmente los *proxies* fueron diseñados para mejorar la eficiencia de las redes más que por razones de seguridad; éstos con conocidos como *caching proxies*, los cuales conservan una copia de la información de cada petición que se realiza en la red. La ventaja de este tipo de *proxies* es que si diez clientes hacen la misma petición en un día, la información puede ser dada directamente por el *proxy*.

La transparencia es la mayor ventaja de los servicios *proxy*. Para el usuario, los *proxies* son como una ilusión de que se están conectando al servidor real. Para el verdadero servidor, el *proxy* le presenta la ilusión de que el usuario está trabajando directamente en el *proxy*.

Los *proxies* son efectivos únicamente cuando son utilizados con un mecanismo que restringe las comunicaciones tanto de salida como la de entrada. Las redes que se encuentran fuera de nuestra zona protegida nunca deben de tener contacto con nuestras máqinas de nuestra red, y las máquinas que se encuentran dentro de nuestra red nunca deben de tener contacto con las máquinas de fuera. Ésta es la función principal de nuestros *proxies*, el tener una barrera entre las dos redes, y cuando desean comunicarse una con otra unicamente lo podrán hacer por medio de nuestros *proxies*, nunca existirá un contacto directo entre ellas.

En la figura 2.3 se muestra la configuración que se necesita para tener un esquema con *proxy*: un cliente *proxy* y un servidor *proxy*. Cuando el cliente desea información desde el verdadero servidor tiene que pasar a través del *proxy*, éste decidirá si sus reglas le permiten o no hacer esta conexión hacia

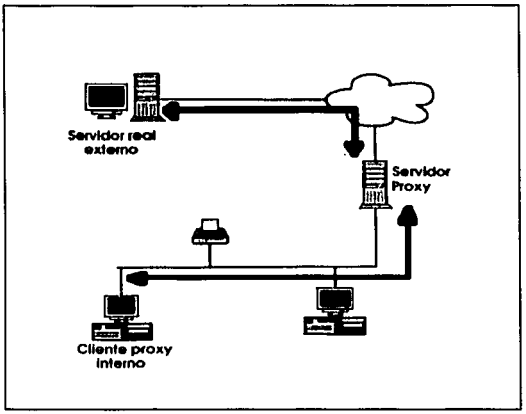


Figura 2.3: Esquema de Proxy

ese servidor. Una vez que se aprobó la petición, el mismo *proxy* es el que realizará la conexión al verdadero servidor como si fuera el *proxy* el que está solicitando la información. Una vez que el servidor le envía de regreso la información al *proxy*, éste deberá realizar el envío de la información al cliente, de manera que el cliente no sepa que hubo de por medio esta evaluación y este intercambio de direcciones.

El tener el *proxy* nos puede ayudar no sólo a controlar las conexiones que realizan nuestros usuarios, también nos dará la posibilidad de guardar en bitácoras las conexiones, y de realizar reglas de forma selectiva del origen/destino de las peticiones que vienen de nuestra red.

Traducción de direcciones

La traducción de direcciones, o mejor conocida como *NAT* (por sus siglas en inglés *Network Address Translation*) permite a una red utilizar un grupo de direcciones de red internamente y uno diferente cuando se comunican con redes externas. La traducción de direcciones no puede por sí misma proveer de seguridad, pero esto ayuda a esconder las direcciones internas y forzar que las conexiones pasen a través de un punto de ahogo (ya que las conexiones

TESIS CON
FALLA DE ORIGEN

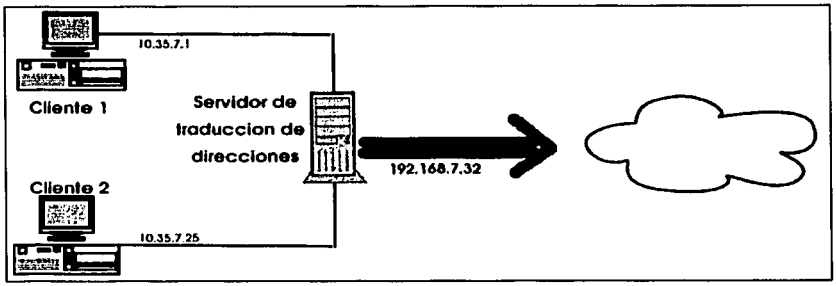


Figura 2.4: Esquema de NAT

sin traducción no podrán trabajar, y el punto de ahogo hará la traducción).

Como en el filtrado de paquetes, la traducción de direcciones de red trabaja teniendo un ruteador haciendo trabajo extra. En este caso no sólo enviará los paquetes, también los modificará. Cuando una máquina interna envía un paquete hacia el exterior, el sistema de traducción de direcciones modifica la dirección de origen para que parezca que el paquete viene de una dirección válida. Cuando una máquina externa envía un paquete al interior de la red protegida, el *NAT* modifica la dirección de destino hacia la dirección interna correspondiente a la petición que se está haciendo. El *NAT* puede también modificar el puerto de origen y destino (esto se le conoce como traducción de dirección de puerto o *PAT*). Esto se puede ver en la figura 2.4.

NAT puede utilizar diferentes esquemas para hacer la traducción entre las redes internas y externas:

- Tener una dirección de red para cada máquina en nuestra red interna y siempre aplicar la misma traducción. Esto no ayuda al ahorro de direcciones, y hace que las conexiones se vuelvan más lentas.
- Proveer dinámicamente una dirección externa cada vez que una máquina interna incia una conexión, sin modificar los números de los puertos. Esto limita el número de máquinas internas que se pueden conectar simultáneamente al exterior.
- Proveer dinámicamente direcciones y puertos externos a nuestra red

TESIS CON
FALLA DE ORIGEN

interna cada vez que el *host* inicia una conexión. Esto hace la forma más eficiente de utilizar direcciones externas.

2.3.2. Sniffers

Los *sniffers* se pueden considerar una de las herramientas más utilizadas por los intrusos, y también una de las más dañinas. Los *sniffers* permiten a los intrusos estar monitoreando todo lo que pasa a través de la red, y al estar espiando esto pueden obtener algunos *passwords* de las conexiones inseguras.

Inicialmente los *sniffers* fueron diseñados para resolver problemas del tráfico de red, se necesitaba monitoreo permanente y así se encontrarían los errores. Éstos esencialmente capturan, interpretan, y guardar la información de los paquetes que pasan a través de la red para un análisis futuro. Esto proveé a los ingenieros de redes de una ventana a la red para poder resolver problemas de modelaje de la red o protocolos.

Dsniff

dsniff es una colección de herramientas para auditoría de redes y pruebas de penetración, *dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf* y *webspy* monitorean la red de forma pasiva buscando informacion interesante (como passwords, correos, archivos, etc.). *arpspoof, dnsspoof*, y *macof* facilitan la intercepción de tráfico de red normalmente indisponible para el atacante (por ejemplo información de la capa 2 de *TCP/IP*). *sshmitm* y *wemmitm* implementan ataques del tipo *monkey-in-the-middle* en contra de sesiones de *ssh* o *https* explotando enlaces débiles.

2.3.3. Analizadores de consumo de banda

Una de los objetivos principales por los cuales se puede intentar penetrar a una red es por el ancho de banda. Normalmente sitios como universidades o el gobierno son lugares muy llamativos para los *hackers* por el gran ancho de banda que tienen disponible.

El analizar el estado del ancho de banda nos ayudará principalmente a saber en qué se está utilizando nuestro medio y por quién. Esto también nos ayudará a saber si tenemos suficientes recursos, en caso contrario podríamos tratar de conseguir un mejor enlace para evitar las coliciones que ocasiona el tener un ancho de banda saturado.

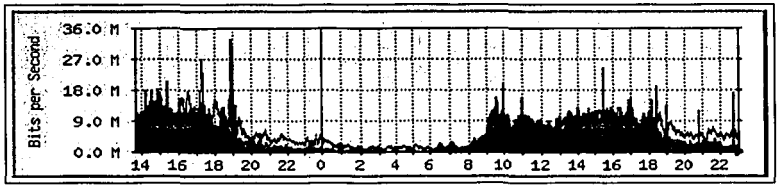


Figura 2.5: Gráfica de MRTG

MRTG

El graficador de tráfico de multidireccionamiento (*MRTG* por sus siglas en inglés) es una herramienta que monitorea el tráfico en las redes. *MRTG* genera código *HTML* que contienen gráficamente la representación instantánea del tráfico de la red. Esto es muy útil para observar el tráfico de entrada/salida en nuestra red de una manera amigable a nuestros ojos. El tipo de gráfica que nos reporta es la mostrada en la figura 2.5.

MRTG consiste de un programa en Perl que utiliza *SNMP* para leer los contadores del tráfico de los ruteadores y un programa en C que realiza el almacenamiento de la información y crea las gráficas representando el monitoreo de las conexiones a la red. Además de una gráfica diaria detallada, *MRTG* también crea representaciones visuales del tráfico visto durante los últimos días, las últimas cinco semanas y los últimos doce meses. Esto es porque *MRTG* no sólo guarda la información actual, sino que crea toda una bitácora de la información que nos puede ser útil.

MRTG no sólo se limita al monitoreo de redes, es posible realizar monitoreo de cualquier variable del *SNMP*. Como se verá más adelante, en esta tesis se utiliza *MRTG* para crear bitácoras de la recepción de virus y correo *spam* en nuestro servidor de correos con un pequeño *script*.

IPFM

El contador de flujo IP (*IPFM* por sus siglas en inglés) es una herramienta de análisis del ancho de banda, que mide qué tanto ancho de banda está utilizando un *host* especificado en sus conexiones a *Internet*. Al ser un programa escrito utilizando las bibliotecas de *libpcap* es totalmente portable a una gran variedad de versiones de *Unix*.

IPFM produce archivos de texto informando el consumo del ancho de

TESIS CON
FALLA DE ORIGEN

Cuadro 2.1: Salida de IPFM

HOST	IN	OUT	TOTAL
host1.domain.com	12345	6666684	6679029
host2.domain.com	1232314	12345	124465

banda por *host* en bytes. El cuadro 2.1 nos muestra la salida de *IPFM*.

Estos datos en la administración de redes son sumamente importantes, con esto podemos saber cuál de nuestros clientes está utilizando una mayor cantidad de ancho de banda, con esto podemos detectar algún tipo de comportamiento extraño en nuestros clientes y detectar a tiempo alguna intrusión.

Otra de las grandes características de *IPFM* es la gran flexibilidad en la configuración, se puenden configurar:

- Clientes de bitácoras
- Intervalo de tiempo de salida
- Archivo de salida
- Ordenamiento por entrada, salida o total

2.3.4. Escaneo de puertos

El escaneo de puertos se utiliza para saber las puertas que están abiertas en una máquina. Como lo hablamos anteriormete, los servicios de *Internet* se basan en puertos que están escuchando en la red, esperando a que algún cliente los llame para responderles. El tener estos puertos abiertos es un gran problema de seguridad, pues todos éstos son puertas a nuestro sistema. Esta herramienta además de decirnos qué puertos están abiertos, me dará información como las máquinas que están funcionando e incluso el sistema operativo con el que trabajan.

Existen diversos escaneos, ahora hablaremos del escaneo más común: el escaneo de puertos. El escaneo de puertos es el proceso de conectarse a un puerto TCP o UDP en el servidor para determinar qué servicios están funcionando para escuchar conexiones. Determiar los puertos que están escuchando

es crítico para poder saber la versión y tipo de sistema operativo. Los servicios que están abiertos y no son utilizados, pueden ser explotados si existe alguna vulnerabilidad. Ahora se hablará de una de las herramientas típicas para realizar el escaneo de puertos.

Nmap

Nmap ("*Network Mapper*") es una herramienta de software libre para la exploración de redes o auditoria de seguridad. Está diseñado para realizar escaneos de redes grandes de una manera rápida, y funciona perfectamente cuando se realiza el escaneo de un solo *host*. *Nmap* utiliza paquetes IP crudos para determinar los *hosts* que están disponibles en una red, qué servicios (puertos) está ofreciendo, qué sistema operativo está utilizando, qué tipo de paquetes/filtros está empleando, y algunas características de sus arquitecturas.

Observermos una salida de este gran programa:

```
[root@fenix root]# nmap 192.168.12.12 -O
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Interesting ports on tesis.unam.mx (192.168.12.12):
(The 1545 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
109/tcp open pop-2
110/tcp open pop-3
143/tcp open imap2
443/tcp open https
993/tcp open imaps
995/tcp open pop3
Remote operating system guess: Linux Kernel 2.4.0 - 2.4.17 (X86)
Uptime 6.116 days (since Thu Oct 10 13:51:16 2002)
Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds
```

Esto es demasiada información de una máquina!!!

Podemos ver que éste es un servidor de correos que acepta conexiones por una gran cantidad de protocolos, como *ssh, pop, http* e *imap*. Además

Nmap nos dio información como el sistema operativo, sabemos que está corriendo una versión del kernel entre la 2.4.0 y la 2.4.17 y que tiene seis días funcionando.

Esta información me sirve para buscar vulnerabilidades en los servicios que ahora sé que está ejecutando, además de que sabiendo el kernel puedo buscar por algún *DoS* o algún *rootkit* que me ayude a obtener la cuenta del administrador en este sistema!!!

2.3.5. Detectores de Intrusos

Los sistemas detectores de intrusos (*IDS*) monitorean los paquetes de las redes y tratan de descubrir si algun hacker/cracker está tratando de entrar a algún sistema (o causar algún ataque DoS). Un ejemplo típico es un sistema que está observando un gran número de peticiones de conexiones TCP a una gran cantidad de puertos a una computadora, como resultado se encuentra que se está realizando un escaneo de puertos TCP. Un *IDS* puede correr incluso en la máquina que se trata de proteger, observando su propio tráfico en la red, o en una máquina independiente que está observando la red de una manera promiscua (como un *hub* o un ruteador).

Capítulo 3

Caso práctico: red del IFUNAM

Actualmente el trabajo con una computadora de escritorio, conectada a la red de la empresa o con *Internet* se ha vuelto cosa de todos los días. El contar con una red confiable es uno de los puntos más importantes cuando la empresa está conectada a la red de redes o simplemente trabaja con información importante.

Esta tesis plantea la solución a problemas de seguridad en la red del Instituto de Física de la Universidad Nacional Autónoma de México (IFUNAM), y pretende explicar el procedimiento que se puede emplear para resolver varios de estos problemas que se pueden presentar en cualquier red.

Dividiremos el análisis de la problemática de seguridad en tres niveles: a nivel de los usuarios, los problemas se observan cuando se trabaja en la PC de escritorio; a nivel de redes cuando los problemas que se encuentran son desde un punto de vista administrativo; y por último a nivel de los servidores.

3.1. Problemática a nivel de usuario

Los usuarios son el mejor monitor para nuestros servicios, ellos son los que nos dan la pauta de lo que necesitan, y el trabajo del administrador es el encontrar el punto medio entre las políticas de seguridad (que incluso pueden sonar exageradas para los usuarios) y el satisfacer todas las necesidades de nuestros usuarios.

Los problemas que puede encontrar el usuario son muchos, y en muchas ocasiones los mismos usuarios son los que provocan esos problemas, el encontrarlos y resolverlos es la tarea de la gente encargada de cómputo, pero

desafortunadamente muchas veces se deja de lado la seguridad, y ése es un punto que nunca se debe de perder de vista.

3.1.1. Inestabilidad de la red

Cuando los usuarios están conectados a *Internet* muchas veces pueden experimentar problemas de inestabilidad. Lo que los usuarios detectan es que aparece el típico mensaje de que la página no pudo ser encontrada, o que tarda demasiado tiempo en bajar una página, o no pueden ver a otras máquinas que se encuentran dentro de su mismo grupo de trabajo.

En la red del IFUNAM encontramos varias redes con grupos de trabajo de Windows en los que pueden compartir archivos, impresoras o cualquier otro recurso informático. Cuando existen problemas de red estos recursos dejan de estar disponibles al no poder conectarse a la red, y esto implica una gran pérdida de tiempo, tanto para los usuarios como para los administradores.

Otro gran problema es cuando se utilizan conexiones remotas, tanto para realizar cálculos en alguna máquina en especial, o para escribir correos o programas de shell o en algún editor de Unix, etc. El tener la red disponible y de manera estable es importante, puesto que al conectarse el usuario a un equipo remoto se necesita tener una conexión continua. Muchas ocasiones las conexiones se cortaban, si se trataba de pasar información de una máquina a otra era exageradamente lento, y por la misma inestabilidad de la red en muchas ocasiones, después de un tiempo considerable de estar transmitiendo la información, la conexión se interrumpía y había que volver a empezar la transmisión, lo que es sumamente molesto cuando se trabaja en una red.

3.1.2. Lentitud de los sistemas

Cuando existen virus en la red o en el sistema, uno de los primeros síntomas es que los usuarios detectan que la máquina se está volviendo muy lenta. Actualmente existen más de 60000 virus diferentes, y este número crece de manera exponencial. Existe una gran variedad de virus, y cada uno de ellos puede hacer cosas diferentes. Ahora existen virus que simplemente se copian de una máquina a otra, no hacen nada en especial más que copiarse a sí mismos, pero esto ocupa una gran cantidad de recursos, como memoria, procesador, disco duro, ancho de banda, etc., y en realidad estos son los virus más benignos.

Existen virus muy agresivos, que además de ocupar un gran ancho de banda pueden hacer que nuestra red sea bloqueada totalmente, por ejemplo, el gusano *Opaserv* es un gusano de red, este gusano se copia a todas las máquinas que estén compartiendo algún archivo o carpeta en la red, si no tiene los permisos correctos. Las máquinas Windows constantemente están mandando señales de *netbios* buscando máquinas con las que pueda compartir archivos o algún periférico, y cuando existe un gusano como éste, las llamadas se hacen más frecuentes e inunda la red para poder encontrar todos los equipos a los que pueda ser copiado.

3.1.3. Falta de direcciones IP

Todos los usuarios quieren tener acceso a la red para poder utilizar los recursos que se encuentran en el Instituto, y también desean tener salida a *Internet* para poder utilizar todas las posibilidades que esta gran red nos da. Pero existe un gran problema: para poder navegar en *Internet* se necesita tener una dirección *IP*. Como se dijo desde el capítulo uno, las direcciones *IP* son un bien finito, existe desde la planeación un error en la cantidad de computadoras que iban a estar conectadas a *Internet* y es por eso que ahora existe una falta de direcciones *IP* a nivel mundial.

Por esta razón es difícil asignarle una dirección a cada máquina en el Instituto. El IFUNAM es uno de los Institutos de investigación con mayor número de computadoras, se tienen cerca de 500 máquinas al servicio de los investigadores, estudiantes y personal en general. Es por esto que no se tienen suficientes direcciones ya que tenemos un segmento completo y la mitad de otro, lo que nos da un total de 381 direcciones *IP*.

Ante este problema se deben de buscar soluciones, ya que los usuarios necesitan tener a su alcance todas las herramientas posibles, los administradores no pueden simplemente decir que no se tienen suficientes direcciones y cruzarse de brazos.

3.2. Problemática a nivel de Red

Uno de los problemas que se presentan en una red insegura es el que la red no esté disponible todo el tiempo, o que cuando uno puede trabajar en la red ésta se vuelva exageradamente lenta. Aquí encontramos que estos problemas se pueden deber a varios factores, como intrusiones en la red, el escaneo de

la red, el tener virus en la red, un mal diseño de red, etc.

3.2.1. Inestabilidad

En el IFUNAM encontramos varios de los problemas de los que hablamos, puesto que se tenía un diseño de red muy malo. El IFUNAM consta de seis edificios, cinco de ellos tenían cableado coaxial y el edificio principal tenía cableado UTP. La conexión entre los edificios es con fibra óptica.

En los edificios con cableado coaxial constantemente se presentaban problemas de pérdida total de conexión a la red. Esto se debe principalmente al tipo de cableado, las redes con cable coaxial son redes viejas e inestables ya que tienen una red tipo anillo, en la que si alguno de los nodos se abre se pierde la comunicación no sólo de ese nodo, sino que toda la red pierde comunicación. Esto se debe a que el cable coaxial trabaja como si fuera un solo cable, todos los nodos dependen de este cable utilizando un conector llamado "T", cuando alguno de los nodos se abre las señales eléctricas se pierden, ya que el circuito se abre, y es por esto que deja de funcionar la red. El cable coaxial tiene otras limitantes, como la gran pérdida de paquetes. las colisiones, la susceptibilidad al ruido electromagnético, etc., por lo que el tener este tipo de cableados ya no era aconsejable.

Hubs

En el edificio principal del IFUNAM se contaba con cableado UTP, este edificio tenía cerca de 210 nodos. En cada piso se encontraban seis *hubs* de 12 puertos cada uno que repartía la señal. El problema con estos dispositivos de red es que comparten el ancho de banda, es decir, que al estar conectados entre ellos escuchan todo el tráfico, no existe una división del tráfico, y si alguno de estos 210 nodos pasaba una gran cantidad de información entre dos computadoras que se encuentran en la misma oficina, ocupaba todo el ancho de banda, por lo que no sólo afectaba a su enlace, si no que todo el *backbone* de 18 *hubs* se volvía lento, por estar compartiendo recursos. Con esto se genera una gran cantidad de colisiones en las transmisiones de red, por estar tan saturado el ancho de banda, por lo que la red era lenta y existía una gran pérdida de paquetes.

Para solucionar esto se tuvo que hacer una reestructuración de la red del edificio principal, se adquirieron *switches*, con grandes beneficios para la red. La primera característica de estos *switches* es que ahora tenían una mayor capacidad, pues los *hubs* anteriores tenían un ancho de banda de 10

MBps, y estos nuevos *switches* tienen un ancho de banda de 100 MBps, por lo que ahora la red es más rápida. Además del beneficio de la velocidad, la tecnología en que se basan los *switches* separa el ancho de banda, por lo que la transmisión entre puertos no implica que la red se deba de volver lenta, pues separa el ancho de banda entre esos puertos y los demás puertos no se ven afectados, por lo que la estabilidad mejoró notablemente.

3.2.2. Tráfico

Uno de los aspectos de la seguridad de redes es la utilización del ancho de banda. Existe una gran cantidad de programas que pueden estar utilizando nuestro ancho de banda de una manera innecesaria, en muchas otras ocasiones ni siquiera se puede dar cuenta el administrador por falta de monitoreo. Cuando vemos una radiografía de lo que está pasando a través de la red nos podemos dar cuenta de que está utilizando nuestro ancho de banda.

La utilización puede ser de cualquier tipo, pero existe un problema cuando es un tipo de utilización no autorizada.

En el IFUNAM se tuvieron varios de estos problemas, y es por esto que se buscó la forma de implementar reglas para que no pueda ser utilizado nuestro ancho de banda con los objetivos que no sean pura y netamente académicos.

Pings

La utilidad *ping* sirve principalmente para saber si un servidor está activo y además podemos calcular el tráfico en la red según el tiempo de su respuesta. Básicamente se le envía un paquete a un servidor y este nos contesta, pero existen algunos ataques basados en este programa, como el *ping* de la muerte. Si se le envía un paquete muy grande al destino puede llegar desordenado, por lo que el servidor pide al origen que le vuelva a enviar una parte o la totalidad del paquete, por lo que se produce un datagrama del *ping* muy grande y producirá su caída. Otra versión de este *DoS* es en la cual la dirección de origen es una dirección falsa, y todas las máquinas a las que les llegue esta petición responderán al sitio falso, lo que hará que ese sitio falso (que normalmente era una organización o alguna página importante) se bloquee por la gran cantidad de peticiones que tiene que responder.

Era muy común tener ataques de *ping* de la muerte que eran lanzados desde fuera del Instituto hacia alguna máquina importante. Se podía observar que se estaba utilizando un gran ancho de banda; al realizar un análisis se vio que estaban utilizando las máquinas del Instituto para enviar este tipo

de ataques a algun sitio en la red. Aquí tenemos dos grandes problemas con nuestros recursos: por un lado están utilizando tiempo de procesador de nuestras máquinas para estar enviando estos paquetes, por otro estaban utilizando nuestro ancho de banda, y eso se ve reflejado en retrasos en la red, lo que hace que nuestros usuarios no la puedan utilizar eficientemente. También es un gran problema hacia afuera de nuestra red, puesto que si los administradores de la red que utilizamos para conectarnos se dan cuenta, pueden incluso bloquear nuestra red para que no se origine un problema mayor.

Este problema se puede solucionar de una manera fácil: bloqueamos todos los paquetes de entrada que son utilizados por el *ping* hacia nuestra red. Esto tiene una gran ventaja: nadie podrá utilizarnos para realizar *ping* de la muerte y también nadie podrá estar escaneando nuestra red para ver si tenemos alguna máquina encendida o no. Pero esto es también un gran problema, puesto que en ocasiones el *ping* es muy útil para saber si alguna máquina está trabajando o no, y al bloquear las entradas del *ping* estaremos bloqueando también este tipo de entradas.

P2P

Actualmente existen un gran número de programas bajo el esquema *Point to Point* (P2P), estos programas basan su funcionamiento en conectar dos computadoras directamente y de esta manera poder establecer pláticas, intercambio de archivos, imágenes, etc. Esto es una gran herramienta que nos da *Internet*, pero de la misma forma estos programas han evolucionado como un gran problema de seguridad en las redes.

Por un lado, estos programas abren puertas que pueden ser explotadas y que pueden dar entrada a un *hacker* por medio de programas que son transmitidos a través de estos protocolos. También se han desarrollado virus para estos programas, por lo que toda nuestra red puede ser infectada por algún virus que fue transmitido por estos programas. También es un gran problema el ancho de banda que utilizan, puesto que el objetivo de muchos de estos programas es el poder compartir archivos, como lo son imágenes o música que puede ser bajada desde cualquier parte del mundo. Los usuarios con el objetivo de poder bajar archivos de la red, también comparten sus archivos, lo que genera un gran tráfico que se ve traducido en lentitud e inestabilidad de las redes. Con esto se debe de plantear la pregunta de qué tan necesario es que los usuarios tengan acceso a este tipo de servicios, y si no es

indispensable la mejor opción es que este tipo de servicios estén bloqueados, y esto se tiene que hacer no sólo por medio del reglamento interno del uso de las redes, también se tiene que forzar por medio del *firewall.*

Virus y gusanos

El gran problema de la actualidad: los virus. Un virus informático es un programa que contiene código que explícitamente se copia y que puede "infectar" otros programas modificándolos o modificando su entorno tanto que el llamar a un programa infectado implica una llamada a la posibilidad de la copia de un virus más evolucionado.

Un gusano (conocido como *worm* por su traducción al inglés) es un programa (o conjunto de programas) autónomo que tiene la posibilidad de difundir copias funcionales de él mismo o de sus segmentos a otras computadoras (usualmente a través de la red). A diferencia de los virus, los gusanos no se necesitan introducirse a un programa de la computadora.

Actualmente, según el CERT (Equipo de respuesta a incidentes), existen más de 62000 virus, y cada día crecen de forma exponencial. Existen una gran cantidad de investigaciones sobre la difusión de los virus, sabemos que se difunden de una manera exageradamente rápida y que necesitamos protección si no queremos tener problemas con ellos. Existen muchas formas de que nuestras computadoras y redes tengan algún problema por virus, con una sola computadora que se encuentre infectada en nuestra red tendremos serios problemas.

Los virus pueden llegar por algún archivo, ya sea que el usuario lo trajo en un floppy, en un CD, que se lo pasaron por correo electrónico o simplemente porque alguna computadora compartió algún archivo infectado. También se pueden transmitir por medio de la red, y para que esto suceda lo único que tiene que hacer el usuario es compartir sus archivos, con lo que el virus podrá entrar.

Para solucionarlo tenemos que establecer políticas de red estrictas respecto a los archivos que se pueden abrir en las computadoras y si se pueden o no compartir archivos en la red y bajo qué reglas. Pero esto no es suficiente, puesto que los virus y gusanos pueden llegar inclusive por el correo electrónico, por lo que se tienen que implementar una gran serie de herramientas para poder mantener a los virus fuera de nuestras redes y máquinas.

Por el lado de los virus que llegan a las máquinas de nuestros usuarios debemos de tener la seguridad de que todas las máquinas tienen un antivirus,

y no sólo eso, sino que ese antivirus debe de ser actualizado constantemente. Para esto se debe de capacitar a los usuarios para que siempre tengan actualizado su antivirus, y concientizarlos que cuando una máquina se infecta no sólo está el problema presente en esa máquina, si no que puede infectar a un gran número de máquinas y archivos en la red.

Para esto se han implementado principalmente tres herramientas en el IFUNAM:

Se consiguieron suficientes licencias de un antivirus comercial el cual es un antivirus sumanente confiable, con esto se garantiza que todas las computadoras del IFUNAM tengan acceso a un antivirus, además de que por ser una institución educativa se pudieron conseguir esas licencias sin costo alguno.

Se instaló una máquina de búsqueda de virus para el servidor de mail. Este motor antivirus, llamado *mailscanner*, es un motor libre, que detiene todos los correos de entrada y salida de los usuarios de nuestro servidor de correos y busca por virus en ellos, en caso de que tengan virus genera reportes para el destinatario, el remitente y el administrador del sistema. Para identificar los virus se utiliza un antivirus libre para linux llamado *F-prot*. Este antivirus es actualizado cada dos horas por medio de un script que baja los archivos de actualización y los instala. Con este esquema podemos asegurarle a nuestros usuarios que todos los correos que envíen y reciban estarán totalmente libres de virus.

Además, después de un incidente de un virus transmitido por un programa del esquema P2P, se cerraron los puertos para la mayoría de todos estos tipos de programas, por lo que estamos libres de recibir este tipo de virus que cada vez se están volviendo más frecuentes en la red.

3.2.3. Robo de direcciones

En la actualidad uno de los grandes problemas de *Internet* es la falta de direcciones *IP*, esto se debe a que el diseño original de la red de redes nunca fue hecho para tener los millones de máquinas que están conectadas actualmente a la red, por lo que este número de direcciones es finito y está muy próximo a agotarse. Esto hace que en cualquier lugar se limiten las direcciones *IP* o que incluso se renten, como lo hacen los proveedores de servicios de *Internet*.

Con este problema a la vista de todos, en el IFUNAM se cuenta con dos segmentos de clase C de estas direcciones, lo que nos da un total de 506

direcciones, pero en el Instituto se tienen cerca de 700 posibles nodos de *Internet* entre computadoras personales, servidores, equipos de redes y de video conferencia, etc., por lo que se tiene un gran déficit en este aspecto.

En algunas ocasiones a los usuarios se les hacía fácil tomar alguna dirección al azar de cualquiera de nuestros segmentos, esto lo que ocasionaba es que se produjeran conflictos de direcciones, puesto que por ninguna razón dos nodos pueden tener la misma dirección *IP*. Lo que esto ocasionaba eran colisiones en la red y que ninguna de las dos máquinas pudiera hacer uso de la red. Esto es grave, pues las colisiones provocan pérdidas de paquetes por lo que nuestra red se vería afectada gravemente en la velocidad y en la confiabilidad, mientras que la desactivación de ambas máquinas es muy grave ya que nuestros usuarios necesitan la red constantemente. Esto es claramente grave a nivel de usuario, pero se puede dar la casualidad que el usuario tomara alguna de las direcciones más importantes de nuestra red, como lo son los servidores o los equipos de red, esto ocasionaría que se perdiera totalmente la red o que se comenzaran a perder correos o accesos a la página puesto que nuestros equipos no están disponibles para contestar a las conexiones.

Para solucionar este problema se tuvo que plantear un esquema de red diferente al que se tenía. Por un lado se hizo una base de datos con la información de todas las máquinas pertenecientes a nuestra red, en la que se incluía la dirección *IP* (en caso de tener una asignada) y la dirección física de la tarjeta de red (dirección *MAC*). Con estos datos se puede bloquear por medio del *firewall* todas aquellas direcciones que no estuvieran asignadas, pero esto no resolvía el problema de cuando un usuario trata de tomar una dirección que ya esté asignada. Para esto se utilizó una herramienta llamada *brconfig*, esta herramienta crea un *bridge* (puente) entre las dos tarjetas de red de nuestro *firewall*, y se le pueden incluir algunas reglas como las direcciones *MAC* que pueden pasar a través de este *bridge*. Con esto garantizamos que sólo se estén utilizando las direcciones *IP* que los administradores de la red estén asignando, y que las máquinas que pueden tener acceso a *Internet* a través de nuestra red, y si algún usuario trata de tomar una dirección que no le corresponde el *firewall* nos reportaría la dirección *MAC* que está tratando de hacerlo, y lo podríamos localizar fácilmente por medio de nuestra base de datos. Si alguna persona que no esté registrada en nuestra base de datos trata de tener acceso a *Internet* por medio de nuestra red no lo podrá hacer, pues el *bridge* no le permitiría la salida.

3.2.4. Barrido de puertos

El *CERT/CC (CERT Corditation Center)* es un centro de desarrollo e investigación federal de los Estados Unidos en materia de seguridad informática, este organismo que se encarga de la respuesta a incidentes informáticos, difusión de cultura informática y búsqueda de vulnerabilidades tiene una página (http://www.cert.org/current/scanning.html) en la que muestra la gran cantidad de escaneos de redes que existen en busca de vulnerabilidades conocidas o para realizar alguna prueba. En el manual de los *hackers* el primer paso es el conocer la red y la máquina que se desea atacar, para esto se realiza un escaneo de puertos sobre todos los nodos que están en la red, en busca de alguna puerta de fácil entrada.

Los *hackers* siempre buscarán el camino fácil para violar la seguridad de una red, como se vio en el capítulo anterior una red será tan fuerte como su eslabón más débil, y el intruso tratará de encontrar ese eslabón, pues será el que le proporcione una menor resistencia.

En un instituto como el IFUNAM, en el que se tienen una gran cantidad de máquinas, y en la mayoría de éstas el usuario es el administrador de la misma máquina, ninguna persona podrá estar revisando constantemente todos los nodos de la red, por lo que tenemos que crear un filtro para todo este tipo de ataques.

Ésta es otra de las razones por la que la instalación de un *firewall* era necesaria, pues con éste se pueden cerrar las puertas de entrada, y dejar abiertos sólo aquellos puertos que sean necesarios, además de que se puede realizar un análisis profundo y dejar que sólo aquellos equipos que necesiten ser accedidos desde fuera de nuestra red tengan los permisos, y los demás equipos estén totalmente bloqueados al acceso exterior, sin perjudicar la salida de nuestros usuarios.

3.3. Problemática a nivel de Servidores

Los servidores son todas aquellas computadoras dedicadas a estar atendiendo peticiones de los usuarios sobre algún o algunos servicios. Anteriormente los servidores tenían que ser máquinas muy poderosas, y éstos eran de alguna tecnología propietaria, por lo que eran muy caros y en muchas ocasiones la administración era sumamente difícil.

3.3.1. Servidores individuales

En el IFUNAM existían una gran cantidad de problemas con los servidores centrales, tanto de seguridad como de disponibilidad para utilizarlos. Por esta razón muchos de los potenciales usuarios de los servidores optaron por poner sus propios servidores, sus servidores personales.

Esto se puede ver como una buena opción, pues así los usuarios podrían tener mayor espacio, tener mejor control sobre sus archivos y así poder asegurar su confidencialidad e incluso el poder ellos tener acceso a servicios que en un servidor público no tendrían.

Pero de la misma forma en que encontramos beneficios a nivel usuario, el tener a su cargo un servidor es una gran tarea, puesto que el tener una máquina con problemas de seguridad puede ser la puerta de entrada a problemas mayores en la red.

Eficiencia de los servidores

Otro gran problema es el relacionado con los ciclos de CPU que se desperdician si tenemos servidores personales. Por definición, un servidor debe de ser una computadora con buenas características, ya que esta computadora estará encendida las 24 horas del día y posiblemente tendrá un gran tráfico, dependiendo de la cantidad de usuarios que tenga y de qué tan frecuentes sean las visitas a nuestro servidor.

Si cada usuario en la red tiene su propio servidor, tendríamos máquinas muy poderosas desperdiciadas por la poca cantidad de uso que le dan los usuarios, o máquinas muy lentas dando servicios que absorven mucho tiempo de procesador, por lo que harían las tareas sumamente lentas e incluso podría tener problemas de estabilidad.

Seguridad en la centralización

Cuando comenzamos este proyecto de seguridad, nos dimos cuenta de que muchos usuarios tenían sus servidores individuales, y esto era un gran problema pues la historia de los servicios de cómputo justificaba que los usuarios tuvieran sus servidores.

Para hacerlos cambiar de opinión se tuvo que hacer una gran campaña de convencimiento, tuvimos que ofrecerles que nuestros servidores iban a ser máquinas dedicadas al 100 % a proveer esos servicios, además de ofrecerles una seguridad mayor a la que ellos tenían (o al menos eso suponemos). Se les explicó que se tendrían administradores especializados en seguridad para

poder tener confidenciales sus documentos e información personal, por lo que tenía una gran cantidad de beneficios el que quitaran sus servidores personales y que nos cedieran las tareas de administración y de servicios a nuestros servidores.

3.3.2. Servidor de correos

El servidor de correos anterior corría sobre una plataforma marca *digital*, con el sistema operativo *Tru64*. Era una máquina un tanto vieja con un sistema operativo viejo, pero actualizado. Aquí encontramos muchos problemas de estabilidad, ademas de que era difícil levantar servicios como *OpenSSH* al ser una tecnología propietaria, si se querían levantar algunos servicios se tenía que contactar al fabricante de estos equipos y comprarles el software especial para esta plataforma. Este es el eterno problema del software propietario.

Para solucionar los problemas del servidor de correos, se optó por comprar una computadora de escritorio potente, actualmente está corriendo en una PC con procesador pentium 4 a 1.8 GHz, 384 MB en RAM y un disco duro SCSI de 40 GB. Esta máquina, sin ser una gran computadora que cueste decenas de miles de pesos, supera por mucho el desempeño de la estación de trabajo *digital*.

Como parte de nuestras soluciones es el empleo de software libre, se le instaló linux, específicamente está corriendo con linux RedHat 7.3. Al tener una máquina que satisface las necesidades de los usuarios podemos promover que nuestros potenciales usuarios utilicen este servidor, pues la seguridad estará totalmente a cargo de nosotros, además de que parte de la ética de un buen administrador es la confidencialidad de los archivos de los usuarios.

Virus

Como se vio durante esta tesis, uno de los grandes problemas de la seguridad informática actual son los virus informáticos, y aunque éstos no afectan a todos los sistemas operativos, sí afecta al sistema operativo utilizado por el 90 % de los usuarios a nivel mundial: Windows. Cuando se le deja la tarea de los antivirus a los usuarios en muchas ocasiones resulta en una mayor pérdida de tiempo, pues al no tenerlo actualizado sus máquinas se infectan constantemente, y con la misma frecuencia están llamando a la gente de soporte en el lugar de trabajo para que les arregle este tipo de problemas.

Con nuestro servidor de correos detuvimos casi totalmente este problema, pues actualmente el 85 % de los virus que afectan a las computadoras de una

empresa llegan por el correo electrónico. Para solucionar esto se instaló un motor de búsqueda de virus en los correos electrónicos llamado *mailscanner*.

mailscanner es un software libre, que al recibir un correo antes de depositarlo en la bandeja de entrada del usuario busca por virus con ayuda de algún antivirus, que en este caso fue f-prot. *mailscanner* es una herramienta con una gran cantidad de configuraciones, actualmente la tenemos buscando virus en todos los correos que entran y salen del instituto, y también permite identificar el correo *spam* con algunas reglas, e inclusive bloquearlo.

Cuando llega un correo con virus a nuestro servidor, *mailscanner* lo detiene y lo aparta, lo analiza con *f-prot* y al comprobar qué tipo de virus tiene lo pone en un directorio para ponerlo en cuarentena. Esto lo hace para que quede una copia del archivo, por si es un archivo sumamente importante se le puede entregar al usuario, bajo su propio riesgo. Cuando encuentra un correo con virus le manda un aviso a la persona que lo envió, indicándole que el archivo que le envió al destinatario tenía un virus, y que por esa razón no fue entregado. De la misma manera le envía un correo al destinatario avisándole que llego un archivo desde el remitente con virus, informa qué virus es el que tiene el archivo y que si lo desea lo puede recuperar avisándole al administrador. Por último, le entrega un correo al administrador avisándole que pone un archivo en cuarentena, que el archivo fue enviado por el remitente al destinatario y que contenía un virus.

Como se puede ver, *mailscanner* es una gran herramienta, nos ayuda a proteger nuestra red y máquinas de virus, y todo esto gracias al software libre y a la centralización que se puede dar cuando los servicios son satisfactorios.

3.3.3. Servidor de Web

Éste era uno de los grandes problemas, pues en el Instituto existía un gran número de servidores web, casi cada investigador que quería tener su información en Internet instalaba su propio servidor web. Ahora hemos puesto a disposición de los investigadores y estudiantes del IFUNAM una muy buena máquina dedicada únicamente a tareas de almacenamiento de páginas para la web.

En este servidor hemos instalado herramientas como *tripwire*. *tripwire* es una herramienta que firma digitalmente los archivos, crea una base de datos de acuerdo al tamaño, fecha de modificación y permisos de los archivos. Esto nos ayuda en la administración, pues aunque tratemos de tener nuestros sistemas al día, debemos de estar preparados en caso de una intrusión. Cuando

alguien *crackea* nuestro sistema, tratará de esconder sus actividades, y una forma de hacerlo es cambiando los binarios del sistema para que en caso de que se monitoré el sistema no podamos ver la actividad de estos intrusos.

Existe el comando en Unix llamado *ps*, este comando nos muestra una lista de los procesos que se están ejecutando en forma real en nuestro sistema. En caso de que algún intruso esté realizando un ataque desde nuestro sistema, en esta lista veríamos los procesos que están ocupando tiempo de nuestro procesador, y de esta manera podríamos detectar el ataque. Pero si nuestro intruso cambia nuestro programa *ps* para que cuando lo ejecutemos no podamos ver los procesos del intruso, no podríamos detectar a este mismo. Es por esto que *tripwire* nos es de gran ayuda, pues al tener una base de datos con la configuración original de nuestro sistema, nos reportaría cualquier cambio de tamaño del binario, y no sólo esto, en la base de datos se guarda información como la última fecha de modificación, y con esto nos avisaría que nuestro binario de *ps* sufrió una modificación, lo cual es absolutamente anormal y podríamos tratar de buscar alguna otra actividad extraña.

Nuestro servidor de web ahora es más seguro, y se encuentra en constante monitoreo, por estos sitios como la Sociedad Mexicana de Cristalografía, la *International Union of Crystallography*, y una gran cantidad de sociedades e investigadores nos han permitido ser un sitio con espejos de sus sitios oficiales, o convertirnos en su sitio oficial.

Estadísticas

Ésta es una parte importante de nuestra seguridad. Ahora tenemos un par de sistemas de estadística para ver cómo se comportan nuestros servidores y nuestros servicios.

Instalamos *webalizer* en los servidores, con esta herramienta podemos ver gráficamente el número de accesos a nuestras páginas, cuáles son las ligas más visitadas e inclusive desde dónde viene el usuario, para de esta forma ver desde qué buscadores o ligas externas nos están conectando más frecuentemente.

Esta información también nos muestra el número de visitas que tenemos diariamente, semanalmente o mensualmente, además de los errores que se han generado, esto nos podría indicar algún intento de violación de la seguridad de nuestros servidores.

En el servidor de correos hemos instalado MRTG, este programa nos

está mostrando gráficamente la cantidad de correos que entran y salen del servidor, y no sólo esto, al tener instalado el antivirus, nos muestra la cantidad de correos con virus que son detectados en nuestro servidor, y acompañando a nuestra campaña de *spam*, nos muestra la cantidad de correos *spam* que está deteniendo.

Como podemos ver el tener estadísticas de nuestros servicios es muy importante, pues podemos darnos cuenta de qué tan eficientes están siendo nuestras herramientas.

3.3.4. Intrusiones

Se dice que existe una intrusión cuando algún atacante puede entrar al sistema y toma el control del mismo. ¿Para qué desearía alguien tener el control de nuestras máquinas? Como lo vimos en el capítulo uno, existen muchas razones por las que un *cracker* desearía tomar el control de nuestros servidores, incluyendo el hacerlo por diversión, el poder tener acceso a la información privada de nuestros usuarios o el querer utilizar nuestro equipo para poder atacar a otros servidores.

Existen muchas tareas que realizar para poder tener un mayor nivel de seguridad en nuestros servidores, y debemos de efectuar cada una de ellas. También es necesario el monitoreo, el monitoreo es algo sumamente importante, desde el punto de vista de esta tesis es más peligroso el tener bitácoras y no revisarlas, que el no tenerlas. Esto porque en las bitácoras se guarda información de los ataques, de correos enviados, recibidos, intentos fallidos de autentificación, etc., todo esto es información que puede ser importante para alguna persona, y es por esto que siempre debemos de tener cuidado con este tipo de información.

Debemos de tener actualizado nuestro sistema para detener este tipo de problemas. Para solucionar esto en el instituto existe una persona dedicada a la seguridad de los servidores, que revisa diariamente por actualizaciones para los servidores, además de encontrarnos suscritos a las listas de correo de seguridad más importantes, de esta manera nos enteramos de las vulnerabilidades y de los parches que salen para nuestros sistemas.

También se deben de instalar herramientas para detectar este tipo de problemas. Cuando un intruso entra a nuestro sistema ejecutará *root kits* los cuales son conjuntos de programas que contienen herramientas para ocultar su intrusión, incluyendo puertas traseras.

Passwords

Un password débil es la puerta principal de entrada para los *crackers*. Debemos de tratar de cambiar constantemente nuestros passwords para cerrar totalmente este problema. Para tratar de evitar este problema en el IFUNAM hemos establecido la política de cambiar passwords cada seis meses, y esto se puede verificar de manera automática con ayuda de *shadow*. *shadow* es uno de los mecanismos de almacenamiento de contraseñas que utilizan los sistemas Unix, y esto nos permite de una manera muy sencilla configurar este tipo de reglas, además de que proveé de una mayor seguridad al sistema pues la característica principal de *shadow* es el evitar que los passwords estén disponibles para que cualquier usuario los pueda ver.

Entrando un poco más a detalle en esto, la base de datos de los usuarios en los sistemas Unix se encuentra en el archivo */etc/passwd*, este archivo tiene información como el nombre, *login*, password, *shell*, directorio de inicio, etc., pero por el diseño original del sistema Unix este archivo tiene permisos de lectura para todo el mundo. Esto es un gran problema, pues después se crearon herramientas para romper passwords, existen herramientas como *John the Ripper* la cual es una herramienta que, al no exister forma de desencriptar los passwords, tiene diccionario y hace pruebas con combinaciones de estos diccionarios para tratar de adivinar alguno de los passwods, después de encriptarlo compara su resultado con el campo correspondiente en el archivo de passwords. Por esta razón se invento *shadow*, el cual ya no guarda los passwords en el archivo original, los guarda en el archivo */etc/shadow*, y este archivo no está disponible para lectura, por lo que no se puede obtener el password encriptado.

También parte importante de esto es la cultura de los usuarios, se hizo una campaña de cambio de passwords, en la que se les pidió a los usuarios que no utilizaran passwords débiles. Se les explicó que un password fuerte tiene las siguientes características:

- Tiene seis caracteres o más.
- No utiliza palabras de diccionario.
- Está compuesto por caracteres alfanuméricos, símbolos y letras mayúsculas y minúsculas.
- Se debe de cambiar al menos cada seis meses.

Capítulo 4

Instalando las herramientas

4.1. Instalando el firewall del IFUNAM

El sistema operativo que se escogió para este firewall fue OpenBSD. A continuación se describirá el proceso que se siguió para instalarlo.

4.1.1. Instalando el Sistema Operativo

Se instaló el Sistema Operativo OpenBSD 3.2 desde el CD. El CD del sistema operativo se puede conseguir por la red en http://www.openbsd.org.

Creando el disco de inicio
La instalación se realizó creando un disco de inicio. El disco de inicio se puede crear a partir de imágenes que vienen contenidas en el disco. Si entramos a la ruta $DISCO/3.2/i386 (en donde $DISCO es la ruta en donde montamos el CD de instalación), encontraremos los siguientes archivos:

- floppy31.fs soporta la mayoría de los sistemas de escritorio, con soporte para las tarjetas de red PCI e ISA, IDE y adaptadores SCSI simples. Soporte para algunas PCMCIA.
- floppyB31.fs soporta varios controladores RAID, por lo que la mayoría de los adaptadores SCSI y varias tarjetas de red EISA e ISA han sido quitados. Éste debe de ser el disco de inicio para varios servidores grandes. Algunos de los adaptadores SCSI menos frecuentemente usados se encuentran aquí.

- floppyC31.fs soporta dispositivos Cardbus y PCMCIA que se encuentran en varias laptops.

- cdrom31.fs soporta la mayoría de los drivers de discos y dispositivos de red de los sistemas instalados; éste es, de hecho, una combinación de los tres discos de inicio anteriores. Se puede utilizar para hacer un floppy de inicio de 2.88 Mb, o más comúnmente, como una imagen de inicio de un CD.

Como nuestro sistema es i386, utilizamos el floppy31.fs, para crearlo utilizamos el siguiente comando:

cd $DISCO/3.2/i386

dd if= floppy31.fs of=/dev/fd0

(estos comandos pueden cambiar dependiendo de la versión de Unix que se esté utilizando, en este caso se utilizó Linux)

La instalación de *OpenBSD* nos llevará por todo el proceso, a continuación sólo se exponen las partes en las que tenemos que responder alguna pregunta.

Comenzamos la instalación

```
erase ^?, werase ^W, kill ^U, intr ^C, status ^T
    (I)nstall, (U)pgrade or (S)hell? i
    Deseamos una instalación
    Proceed with install? [n] y
    Specify terminal type [vt220]: Enter
    Do you wish to select a keyboard encoding table? [n]
```

Aquí puede variar dependiendo de la arquitectura. En este caso (i386) damos enter.

Ahora sigue particionar los discos. OpenBSD tiene una forma muy especial de trabajar con los sistemas de archivos. Primero tenemos que crear una partición para OpenBSD, y después dentro de ésta, OpenBSD crea sus propias particiones y sistemas de archivos.

```
    Which disk is the root disk? [wd0] >Enter<
    Do you want to use the *entire* disk for OpenBSD? [no] yes
```

El *root disk* es el disco donde se encontrará el sistema de archivos / y el *swap*. *wd0* especifica un disco IDE, *sd0* especifica un disco SCSI. En este caso se escogió el disco entero para el sistema, ya que será un servidor y no tendrá ningún otro sistema operativo.

Ahora haremos las particiones para OpenBSD y le pondremos las etiquetas. Para esto se utiliza *disklevel*, parte del programa de instalación.

```
>p
16 partitions:
# size offset fstype [fsize bsize cpg]
a: 12643092 63 unused 0 0
c: 12658776 0 unused 0 0
>d a
>a a
offset: [63] >Enter<
size: [12658713] 3000m
Rounding to nearest cylinder: 163422
FS type: [4.2BSD] >Enter<
mount point: [none] /
>a b
offset: [163485] >Enter<
size: [12495291] 1000m
Rounding to nearest cylinder: 614250
FS type: [swap] >Enter<
>a d
offset: [777735] >Enter<
size: [11881041] 200m
Rounding to nearest cylinder: 163485
FS type: [4.2BSD] >Enter<
mount point: [none] /tmp
>a e
offset: [941220] >Enter<
size: [11717556] 2000m
Rounding to nearest cylinder: 163485
FS type: [4.2BSD] >Enter<
mount point: [none] /var
>w
>q
```

Aquí se pueden observar las particiones originales del disco duro. Con el comando *d a* eliminamos la partición a. En OpenBSD siempre existe la partición *c*, que es la que hace referencia a todo el disco.

Lo siguiente fue agregar la partición *a (a a)*, offset es el valor del primer sector que tomará esta partición, a continuación dimos el tamaño de la partición. El prefijo *m* es para indicar que el valor que estamos dando es en megabytes.

Nos informa que tomó hasta el cilindro 163422 y a continuación definimos el tipo de sistema de archivos que tendrá la partición. En este caso será 4.2BSD (nativo de OpenBSD). Por último, definimos que el punto de montaje de esta partición será / (la raíz).

De la misma forma agregamos las particiones *b*, *d* y *e*. Para la partición *b* definimos el sistema de archivo como *swap*. Para las particiones *d* y *e* definimos el sistema de archivos como *4.2BSD*, pero definimos los puntos de montaje como */tmp* y */var* respectivamente.

```
The following partitions will be used for the root
filesystem and swap:
wd0a /
wd0b swap
Mount point for wd0d (size=81742k) [/tmp, RET, none, or
done]? >Enter<
Mount point for wd0e (size=81742k) [/var, RET, none, or
done]? >Enter<
Now you can select another disk to initialize. (Do not
re-select a disk you have already entered information for).
Available disks are:
wd0
Which one? [done] >Enter<
```

Aquí nos confirma que los puntos de montaje sean los correctos. Si quisiéramos cambiar alguno aquí lo podemos hacer, de lo contrario sólo confirmamos los datos.

A continuación nos preguntará para confirmar que todos los datos serán borrados al hacer el formateo de las particiones, como se muestra a continuación:

```
================================================================
The next step will overwrite any existing data on:
wd0a wd0h wd0d wd0g wd0e
Are you really sure that you're ready to proceed? [n] y
Creating filesystems...
```

Configuración de la red

```
You will now be given the opportunity to configure the network.
This will be useful if you need to transfer the installation
sets via FTP, HTTP, or NFS.
Even if you choose not to transfer installation sets that
way, this information will be preserved and copied into
the new root filesystem.
Configure the network? [y] >Enter<
Enter system hostname (short form, e.g. 'foo'): [] firewall
If any interfaces will be configured using a DHCP server
it is recommended that you do not enter a DNS domain name,
a default route, or any name servers.
Enter DNS domain name (e.g. 'bar.com'): [] fisica.unam.mx
You may configure the following network interfaces (the
interfaces marked with [X] have been successfully configured):
[ ] fxp0
Configure which interface? (or 'done') [fxp0] >Enter<
IP address (or 'dhcp')? [] 132.248.7.1
Symbolic (host) name? [firewall] >Enter<
Netmask? [255.255.255.0] >Enter<
Your use of the network interface may require non-default
media directives. The default media is:
media: Ethernet autoselect (none)
This is a list of supported media:
media none
media 10baseT
media 10baseT mediaopt full-duplex
media 100baseTX
media 100baseTX mediaopt full-duplex
media autoselect
media none instance 1
media 100baseTX instance 1
media 100baseTX mediaopt full-duplex instance 1
media 100baseT4 instance 1
If the default is not satisfactory, and you wish to use
another media, copy that line from above (e.g. "media
100baseTX ")
Media directives? [] >Enter<
```

```
You may configure the following network interfaces (the
interfaces marked with [X] have been successfully
configured):
[X] fxp0
Configure which interface? (or 'done') [done] >Enter<
Enter IP address of default route: [none] 132.248.7.254
Enter IP address of primary nameserver: [none] 132.248.204.1
Would you like to use the nameserver now? [y] >Enter<
```

Aquí vemos cómo configuramos nuestro sistema para que tenga una dirección estática. Podríamos configurarlo para que obtuviera la dirección por DHCP y la información de ruteo y DNS la obtendría directamente del servidor.

Escogiendo el medio de instalación

```
Escape to shell? [n] >Enter<
/dev/wd0a on /mnt type ffs (rw, asynchronous, local,
ctime=Tue Apr 16 15:50:28 2002)
/dev/wd0h on /mnt/home type ffs (rw, asynchronous,
local, ctime=Tue Apr 16 15:50:28 2002)
/dev/wd0d on /mnt/tmp type ffs (rw, asynchronous,
local, ctime=Tue Apr 16 15:50:28 2002)
/dev/wd0g on /mnt/usr type ffs (rw, asynchronous,
local, ctime=Tue Apr 16 15:50:28 2002)
/dev/wd0e on /mnt/var type ffs (rw, asynchronous,
/local, ctime=Tue Apr 16 15:50:28 2002)
Please enter the initial password that the root
account will have.
Password (will not echo): >enter root PW<
Password (again): >re-enter root PW<
Do you expect to run the X Window System? [y] n >Enter<
You must now specify where the install sets you want to use
are. They must either be on a local device (disk, tape,
or CD-ROM), an accessible NFS filesystem or an accessible ftp
or http network server. You will have the chance to repeat
this step or to extract sets from several places, so you do
not have to try to load all the sets in one try and can
recover from some errors.
Install from (f)tp, (h)ttp, (t)ape, (C)D-ROM or local
```

```
(d)isk? C
The following CD-ROM devices are installed on your system.
Please make sure the CD is in the CD-ROM drive and select
the device containing the CD with the installation sets:
cd0
Which CD-ROM contains the installation media? [cd0] >Enter<
Enter the directory relative to the mount point that
contains the file: [3.1/i386] >Enter<
```

Ahora tendremos que escoger la forma de instalación. Primero nos pregunta si queremos un shell, esto es para hacer algunos cambios en forma manual a los archivos que se han ido creando en el disco. Nos pide el *password* de root. Es importante notar que no tiene eco, es decir, no se mandan los caracteres que pongamos en el teclado a la pantalla.

Después nos pregunta si esperamos correr X en nuestro sistema. X es el ambiente gráfico, y por seguridad no debemos de levantar el ambiente gráfico en servidores, pues esto abriría puertos innecesarios. La siguiente información que nos pregunta es acerca de desde dónde vamos a instalar. En este caso instalaremos desde el CDROM, por lo que le damos la opción C. La siguiente pregunta es referente a la unidad de CD que tomará. En nuestro caso sólo tenemos una, por lo que escogeremos (dando enter para tomar el default) cd0. Por último nos pregunta sobre el directorio de donde tomará la información. El default de esta pregunta tiene el directorio correcto, por lo que un enter lo tomará.

Escogiendo los paquetes

Ahora escogeremos los paquetes que deseamos instalar. En esto tenemos que ser muy cuidadosos, ya que éste será un *firewall*, debemos de tener cuidado de no escoger paquetes que no sean necesarios. Entre más cosas innecesarias tengamos, la posibilidad de algún error en el software es mayor.

```
The following sets are available. Enter a filename, 'all'
to select all the sets, or 'done'. You may de-select a set
by prepending a '-' to its name.
[X] base31.tgz
[X] etc31.tgz
[X] misc31.tgz
[X] comp31.tgz
[X] man31.tgz
```

```
[ ] game31.tgz
[ ] xbase31.tgz
[ ] xshare31.tgz
[ ] xfont31.tgz
[ ] xserv31.tgz
[X] bsd
File name? [xbase31.tgz]
```

El *prompt* nos pregunta por más paquetes. Si quisiéramos instalar los juegos, por ejemplo, tendríamos que dar la opción +game31.tgz.

Podemos instalar diferentes combinaciones de paquetes, los cuales contienen lo siguiente:

base31.tgz Contiene el sistema base OpenBSD Requerido

etc31.tgz Tiene todos los archivos del /etc Requerido

comp31.tgz Contiene el compilador y sus herramientas, bibliotecas. Recomendado

man31.tgz Páginas del manual Recomendado

misc31.tgz Contiene la información miscelánea, documentos de configuración.

game31.tgz Los juegos para OpenBSD

xbase31.tgz Instalación base para X11

xfont31.tgz Contiene el servidor de fuentes de X11 y las fuentes

xserv31.tgz Tiene los servidores de X11

xshare31.tgz Contiene las páginas del manual, configuraciones locales, etc. Para X.

bsd El Kernel. Requerido

Los paquetes que hemos escogido son suficientes para nuestra tarea de filtrado.

```
Ready to install sets? [y] >Enter<
```

La siguiente pregunta que nos hará es referente a la zona horaria. Lo único que tenemos que dar es la opción de México/General para que tome nuestra zona horaria.

Hemos terminado con la instalación del sistema operativo, ahora sigamos con la configuración del firewall.

4.1.2. Configurando el firewall

Primero revisaremos que nuestros parámetros de red sean los correctos. Para esto utilizamos el comando *ifconfig*.

```
lo0: flags=8009<UP,LOOPBACK,MULTICAST>mtu 33224
inet6 fe80::1%lo0 prefixlen 64 scopeid 0x6
inet6 ::1 prefixlen 128
inet 127.0.0.1 netmask 0xff000000
lo1: flags=8008<LOOPBACK,MULTICAST>mtu 33224
xl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,
MULTICAST>mtu 1500
media: Ethernet autoselect (100baseTX full-duplex)
status: active
inet6 fe80::204:76ff:fed1:78f5%xl0 prefixlen 64 scopeid 0x1
xl1: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,
MULTICAST>mtu 1500
media: Ethernet autoselect (100baseTX full-duplex)
status: active
inet6 fe80::204:76ff:fed1:8779%xl1 prefixlen 64 scopeid 0x2
pflog0: flags=141<UP,RUNNING,PROMISC>mtu 33224
sl0: flags=c010<POINTOPOINT,LINK2,MULTICAST>mtu 296
sl1: flags=c010<POINTOPOINT,LINK2,MULTICAST>mtu 296
ppp0: flags=8010<POINTOPOINT,MULTICAST>mtu 1500
ppp1: flags=8010<POINTOPOINT,MULTICAST>mtu 1500
tun0: flags=10<POINTOPOINT>mtu 3000
tun1: flags=10<POINTOPOINT>mtu 3000
enc0: flags=0<>mtu 1536
bridge0: flags=41<UP,RUNNING>mtu 1500
bridge1: flags=0<>mtu 1500
vlan0: flags=0<>mtu 1500
vlan1: flags=0<>mtu 1500
gre0: flags=8010<POINTOPOINT,MULTICAST>mtu 1450
gif0: flags=8010<POINTOPOINT,MULTICAST>mtu 1280
```

Aquí tenemos mucha información útil, pero por el momento sólo nos fijaremos en las tarjetas de red. Éstas están identificadas (en este caso, siempre dependen del tipo de tarjeta y tecnología que utilicemos) como xl0 y xl1.

Verificamos que estas tarjetas no tienen dirección. Esto es para hacer un firewall transparente. Con esto queremos decir que el filtrado será totalmente transparente para los usuarios, ellos no tendrán que configurar absolutamente nada en sus máquinas. Esto es muy útil ya que muchas veces resulta latoso el tener que cambiar la configuración de 500 máquinas.

Estas tarjetas se pueden configurar fácilmente en sus respectivos archivos, a continuación se muestran los archivos de configuración de las tarjetas

```
Fire# cat /etc/hostname.xl0
media autoselect up
Fire# cat /etc/hostname.xl1
media autoselect up
```

Lo siguiente es configurar nuestra máquina para que permita hacer el *forward*, que es una configuración necesaria para poder filtrar por medio de las direcciones MAC o por medio de paquetes. Para hacer esto tenemos que hacer un cambio en el archivo /etc/sysctl.conf. Para esto, tenemos que agregar la siguiente línea al archivo de configuración:

```
net.inet.ip.forwarding=1
```

Creando el bridge

Necesitamos crear un *bridge* entre las tarjetas de red. Este *bridge* nos permitirá filtrar las direcciones MAC. Las direcciones MAC son las direcciones físicas de las tarjetas de red. Todas las tarjetas de red tienen una dirección única asociada. Lo que pretendemos con esto es tener un control de todas las máquinas que se conectan a la red del Instituto. El protocolo TCP/IP (por el cuál funcionan todas las conexiones a Internet) tiene aún muchos problemas de seguridad, entre ellos que cualquier máquina pueda tomar la dirección IP de otra máquina. Nosotros tratamos de evitar esto haciendo un filtro por la dirección de cada tarjeta, de esta forma, si alguna máquina toma una dirección que no le corresponde, tendríamos un problema que nos sería reportado por nuestro sistema de monitoreo.

La utilería *brconfig* nos muestra el estado del kernel sobre las interfaces de *bridge* y permite al usuario el control de estos *bridges*. Los dispositivos de tipo *bridge* crean una liga lógica entre dos o más interfaces *ethernet*, el cual mandará de forma selectiva cada uno de los paquetes que lleguen a nuestro *bridge* hacia algún otro *bridge*. Éste se puede utilizar para separar el tráfico entre grupos de máquinas distintas en el mismo segmento y proveer filtrado transparente a los datagramas de IP.

Las reglas harán que las máquinas que no tienen permiso de salir, por no estar en nuestra base de datos, no podrán salir. A continuación se muestra un ejemplo de cómo se deben de construir las reglas. Le debemos de dar la dirección MAC a la cual le estamos dando permiso de salir de nuestra red, y la acción que debe de tomar. En este caso, debemos de darle la acción *pass*,

para que le permita salir. Es importante anotar que la última línea de este ejemplo es *block in on xl1*.

Esta línea es para negar la salida a cualquier otra tarjeta que no haya encontrado en la lista. *brconfig* funciona de manera secuencial, si encuentra la regla de salida primero, con ésta se queda y permite la salida. Si después de haber revisado todas las reglas y de no haber encontrado la dirección que le está pidiendo salir, toma la línea del final, que es el default, y por esta razón bloquea la salida. A este archivo se le puso el nombre de */etc/bridgename.bridge0.rules*, así se referirá a él.

```
pass in on xl1 src 00:60:08:CE:F6:DA
pass in on xl1 src 00:60:08:CE:F3:87
pass in on xl1 src 00:30:65:4E:BE:72
pass in on xl1 src 00:01:02:C9:41:A8
pass in on xl1 src 00:C0:F0:1F:6B:1A
pass in on xl1 src 00:60:97:C9:3D:97
pass in on xl1 src 00:50:DA:5E:42:14
pass in on xl1 src 00:60:08:3D:A0:43
pass in on xl1 src 00:50:BF:7A:AA:89
pass in on xl1 src 00:00:E8:98:D8:19
pass in on xl1 src 00:40:05:39:10:4C
pass in on xl1 src 00:00:39:22:CB:84
pass in on xl1 src 00:04:76:D1:87:5A
pass in on xl1 src 00:50:BF:63:EA:28
pass in on xl1 src 00:60:08:3D:9F:46
pass in on xl1 src 00:20:AF:70:F2:C9
pass in on xl1 src 00:50:BF:61:0C:9E
pass in on xl1 src 00:10:B5:CD:6C:78
pass in on xl1 src 00:50:BF:63:F3:C4
pass in on xl1 src 00:10:B5:CD:6B:93
pass in on xl1 src 00:50:BF:60:EB:DC
pass in on xl1 src 00:10:83:41:AA:D0
pass in on xl1 src 00:90:04:39:21:78
pass in on xl1 src 00:60:08:F7:17:C8
block in on xl1
```

Ahora que ya tenemos el archivo de configuración podemos crear el *bridge* y activar las reglas. Para crear un *bridge* se utiliza el siguiente comando:

```
Fire# brconfig bridge0 add xl1 up
```

Y para activar las reglas del *bridge* lo hacemos con el siguiente comando:

```
Fire# brconfig bridge0 rulefile /etc/bridgename.bridge0.rules
```

4.1.3. Creando las reglas de filtrado de paquetes

Existen muchos documentos interesantes sobre el filtrado de paquetes. El filtrado de paquetes se basa en los protocolos de TCP/IP que permitirá el *firewall* entrar y salir. Estas reglas cambian constantemente, dependiendo de las necesidades de los usuarios y de lo que se desea bloquear. El filtrado lo hacemos con una herramienta que tiene *OpenBSD* llamada *Packet Filter* (*pf*). *pf* nos permitirá bloquear por protocolos, por puertos, paquetes de entrada, paquetes de salida, destino, origen, hacer *logs*, etc. Es una herramienta muy poderosa.

Inicialmente creamos reglas para que los usuarios del Instituto pudieran salir con cualquier protocolo y hacia cualquier lugar de Internet sin limitaciones. Hemos ido ajustando poco a poco las reglas de entrada, que son las más importantes ya que no conocemos a las personas que se quieren conectar a nuestra red. Bloqueamos todos los *pings* que vienen hacia el Instituto.

Es muy difícil de explicar porqué hemos ido torneando las reglas de esta manera, por ejemplo, los *pings* los bloqueamos porque habíamos tenido ataques llamados *ping de la muerte*, en la que una máquina mandaba muchísimos *ping* hacia un *site* para poder bloquearlo. Este tipo de ataques no es difícil hacerlo, y el protocolo lo permite. Poco a poco seguiremos avanzando con nuestras reglas para hacer el filtrado. En el anexo A se muestra la configuración base del archivo */etc/pf.conf* que es el archivo de configuración para el filtrado de paquetes. Todas las líneas vienen comentadas para su mejor comprensión.

Las reglas tienen un formato parecido al del *bridge*. Se van leyendo de forma secuencial, y se queda con la última que coincidió. Tenemos varias instrucciones que nos pueden ayudar con la configuración de las reglas. La forma de trabajar de *pf* nos puede confundir, pero es importante el entender que *pf* sigue leyendo hasta el final del archivo, a menos que le demos la orden para que se detenga en esa regla. Si nosotros creamos nuestras reglas y al final ponemos la instrucción *block in log all*, lo haríamos esperando que bloqueara todo, pero si en algún lugar ponemos la orden *quick* (para que se salga en esa regla), se quedará en esa última regla. A continuación muestro un pequeño cuadro que facilitará la comprensión de las reglas:

pass Permite que el paquete entre/salga

block Bloquea el paquete

all Se aplica para todos los paquetes
log Crea bitácoras (*logs*) de los paquetes que coincidan con esa regla
quick Aplica la regla y no continúa leyendo las demás
proto Definimos el protocolo del que estamos hablando
port Para aplicar a algunos puertos en especial

Ya que tenemos configurado nuestro archivo de reglas, y ahora que deseamos probarlo, nos falta ver cómo activamos estas reglas. Debemos de tener mucho cuidado con esto cuando estamos trabajando de manera remota, pues tal vez alguna de nuestras reglas se puede aplicar para nuestra conexión actual. Se recomienda que cuando se desee actualizar las reglas, o reiniciar las actuales, se hagan en la consola por si existe algún problema. La forma de actualizar (o de iniciar) las reglas es con el siguiente comando:

Fire# pfctl R /etc/pf.conf

¡¡¡Y de esta manera tenemos funcionando nuestro firewall!!!

Para poder ver lo que está pasando en este momento en los logs, lo hacemos con el comando:

Fire# tcpdump i pflog0

Las bitácoras se guardan bajo el directorio /var/log, y las podemos ver con *tcpdump*.

Referencias

http://www.openbsd.org/
http://www.openbsd.org/faq/faq4.html
http://www.openbsd.org/faq/faq6.html
http://www.openbsd.org/cgi-bin/man.cgi
http://www.openbsd.org.mx/
http://www.openbsd.org.mx/~alex/openbsd_despues_de_la_inst/index.html
http://www.openbsd.org.mx/~alex/conf_fw_openbsd/index.html
http://www.openbsd.org.mx/~alex/SS_OpenBSD/index.html
http://www.securityfocus.com/
http://online.securityfocus.com/infocus/1182
http://www.jmu.edu/computing/info-security/engineering/proj/fw/personal.shtml
http://www.deadly.org/pf-howto/html/
http://www.cert.org/
Elizabeth D. Zwicky. Building Internet Firewalls (2nd Edition). Ed. O´Reilly
Wes Sonnenreich. Building Linux and OpenBSD Firewalls. Ed. John Wiley & Sons, Inc

4.2. Instalando el servidor de correos

4.2.1. Instalación

Existe una gran variedad de opciones para instalar un servidor de correos. Tenemos mucho de donde escoger, tanto en hardware como en software. En el caso del hardware no teníamos muchas opciones puesto que teníamos que adaptarnos a lo que estaba disponible en el Instituto.

Tenemos una PC marca Gateway con procesador Pentium IV a 1.8 GHz y 384 Mb en memoria RAM con un disco SCSI de 40 Gb como servidor de correo. Este equipo es muy bueno, aunque no siempre es lo mejor tener una PC como servidor.

Respecto al software tenemos más opciones, pero primero tenemos que analizar para qué se va a utilizar nuestro servidor y de esta manera poder adoptar ciertas tendencias.

Éste será un servidor para cerca de 500 usuarios. Es un servidor de correos principalmente, y necesitamos ciertas herramientas para la facilidad de los usuarios. Este servidor deberá de contar con una interfaz web para que los usuarios puedan revisar sus correos por medio de cualquier navegador. Deberán de poder consultar sus correos por medio de clientes de correo que corran bajo Windows como Outlook y Eudora. Este servidor deberá de contar con un software antivirus para analizar los correos que entran y salen, para de esta manera poder disminuir el gran problema de virus que tenemos en el Instituto. Debe de ser un servidor seguro que cuente con herramientas de encriptación para poder asegurar la información de los usuarios.

En el sistema operativo tenemos básicamente dos opciones de software libre: linux y BSD. Escogimos RedHat Linux 7.3 por las facilidades que tiene linux para la configuración del software solicitado. Linux ha demostrado ser un sistema operativo robusto y con gran soporte.

Buscando en la red encontramos un gran número de software que puede ejecutarse bajo la plataforma de Linux y que sería de gran utilidad para el Instituto. Estas herramientas incluyen antivirus (F-prot), buscador de virus para correo electrónico (mailscanner), conexiones seguras (OpenSSH), cliente de correo para el web (null webmail), encriptación de páginas web (ssl y https), servidor de web (apache), servidor de mail (sendmail), etc, y demás herramientas que nos serán sumamente útiles para brindar un servicio satisfactorio a nuestros usuarios, sin dejar a un lado la seguridad.

Existe una gran cantidad de páginas que nos pueden ayudar a tener una

instalación segura de nuestro Linux, aquí pongo dos que me parecen bastante buenas:

http://www.redhat.com/docs/manuals/linux/RHL-7.2-Manual/

http://www.gwolf.cx/seguridad/recom_seg_linux/

Estas guías nos pueden ayudar a instalar nuestro sistema operativo, aquí se tratará de orientarse a la parte de las configuraciones que existen en especial en nuestro servidor de correos.

Se supone que con las guías anteriores ya tenemos instalado nuestro sistema operativo de la forma en que nos dicen en la guía, y hemos tomado algunas medidas de seguridad como las que vienen en las mismas páginas.

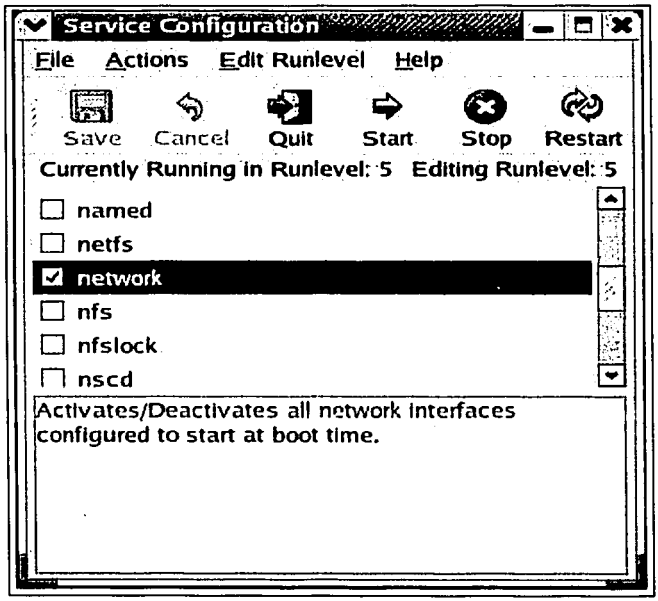
4.2.2. Configurando los servicios del sistema

Es muy importante el tener arriba unicamente los servicios necesarios. La instalación de RedHat es insegura por default, cuando se inicia tiene un gran número de servicios innecesarios abiertos, por lo que es necesario que el administrador se ponga a cerrar todas aquellas puertas que no se necesitan tener abiertas. Existen muchas formas de hacerlo. Se pueden editar directamente los *rc.** que se encuentran bajo el directorio */etc/rc.d* o entrando a *system services* de la herramienta *setup* o con la herramienta gráfica *serviceconf*. La herramienta *serviceconf* nos ayudará de una manera gráfica a activar y desactivar todos los servicios. La figura 4.1 muestra la herramienta.

Con esta herramienta podemos observar qué hace cada uno de los servicios que vienen incluidos en esta distribución de Linux, pero para satisfacer nuestros requerimientos nosotros unicamente necesitaremos los mostrados en el cuadro 4.1.

Podemos comprobar las puertas que están abiertas en nuestro sistema utilizando la herramienta *nmap*. *nmap* es una herramienta de software libre para hacer un barrido de puertos. En un ataque, lo primero que hace la máquina atacante es buscar los puertos abiertos en nuestro equipo, para esto utiliza diversas herramientas para recorrer todos los puertos del sistema y buscar el que puede ser vulnerable. Esto puede ser detectado por diversas herramientas de seguridad que debemos de instalar como *portsentry*. El tener los puertos cerrados es sumamente importante, por lo que la salida que nos dé *nmap* como administradores será de gran ayuda. Al ejecutar *nmap* sobre nuestro servidor obtenemos la siguiente salida:

Figura 4.1: Serviceconf



TESIS CON
FALLA DE ORIGEN

Cuadro 4.1: Servicios requeridos para nuestros servicios

Servicio	Descripción
crond	Calendarización de programas
gpm	Soporte para mouse en ambientes de texto
httpd	Apache es un servidor de web
imap	Conexiones con clientes *IMAP.*
imaps	Conexiones con clientes *IMAP* utilizando *SSL.*
ipop2	Conexiones con clientes POP2.
ipop3	Conexiones con clientes POP3.
keytable	Lee el mapa de teclado seleccionado.
network	Activa/desactiva las interfaces de red
pop3s	Conexiones con clientes POP3 utilizando SSL.
sendmail	*MTA* que mueve correos de una máquina a otra.
sshd	Demonio del servidor de OpenSSH
syslog	Demonio para guardar *logs* en bitácoras
xinetd	Poderosa herramienta para limitar servicios

```
[javo@pegaso javo]$ nmap 132.248.7.40
Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
Interesting ports on fisica.unam.mx (132.248.7.40):
(The 1525 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
109/tcp open pop-2
110/tcp open pop-3
143/tcp open imap2
443/tcp open https
993/tcp open imaps
995/tcp open pop3s
Nmap run completed -- 1 IP address (1 host up) scanned in
1 second
```

4.2.3. Parches

Hemos dado el primer paso en la seguridad, pero la tarea del administrador es grande. Una de las tareas más importantes es el tener al día nuestro sistema respecto a los parches y actualizaciones. Tenemos que recordar que el software lo escriben humanos y que es susceptible a errores. Los parches son actualizaciones, en ocasiones de seguridad y en otras de sistema, que se aplican al software que tenemos instalado en nuestras computadoras.

Existen muchas maneras de instalar estos parches, se puede conseguir el código fuente y aplicarlo directamente. Después configurar, compilar e instalar. Uno de los beneficios que tiene RedHat es un software especial para instalar y actualizar software. Este programa se llama *RPM Package Manager* (*rpm*). *rpm* nos permite actualizar de una manera sencilla todo el software que soporta RedHat. Tenemos que estar atentos a las actualizaciones de RedHat.

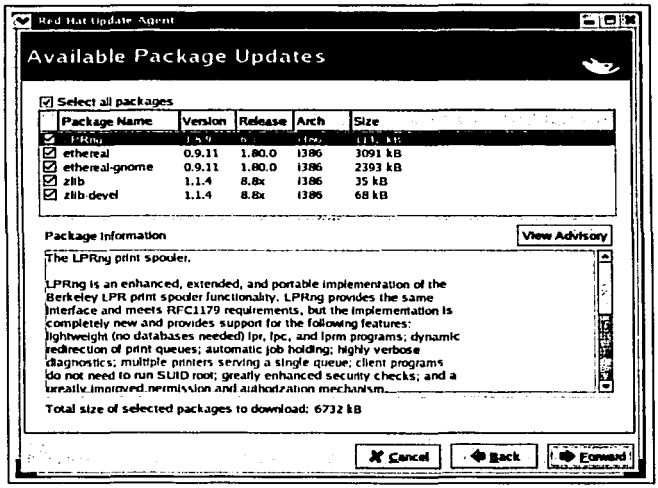
Una vez que los *hackers* saben qué puertos tenemos abiertos en nuestros sistemas, el siguiente paso es identificar las versiones del software que pueden ser vulnerables. Existen programas que hacen las dos cosas al mismo tiempo, como *snort*. Cuando este programa está realizando el escaneo de puertos, realiza conexiones para investigar la versión del software que está abriendo ese puerto, y busca en su base de datos por vulnerabilidades conocidas para esa versión. Es por eso que debemos de tener al día todo el software que tengamos instalado en nuestro equipo, de esta manera estaremos preparados para este tipo de ataques.

Existe un software llamado *up2date* que distribuye RedHat. Este software necesita que uno se registre en RedHat Network, pero facilitará aún más la instalación de los parches. Con este software lo único que tenemos que hacer es ejecutarlo y él solo revisará por nuevas actualizaciones, en caso de existir las baja y las instala. Es una herramienta muy útil y fácil de usar. La figura 4.2 muestra la interfaz.

4.2.4. Antivirus

Ahora que ya tenemos una seguridad media en nuestro sistema, tenemos que empezar a instalar todas las herramientas que necesitamos. La primera que instalaremos será el antivirus. *f-prot* es una máquina antivirus. La máquina antivirus es la encargada de tener las definiciones de los virus y encontrar el código del virus en el archivo que lo contiene. Este software es

Figura 4.2: Up2date



TESIS CON
FALLA DE ORIGEN

libre bajo licencia de *F-Secure inc.*

Para instalarlo lo puedo conseguir del sitio de f-prot, lo puedo bajar con el siguiente comando.

```
cd /home/javo/Antivirus/F-prot
wget "ftp://ftp.f-prot.com/pub/linux/fp-linux-sb.tar.gz"
```

Ahora seguiremos con la instalación. Lo pongo bajo el directorio /usr/local para poder trabajar bajo esta ruta.

```
cd /usr/local
tar -zxvf /home/javo/Antivirus/F-prot/fp-linux_sb.tar.gz
```

Creamos un par de ligas para tenerlas bajo los archivos de ejecución. De esta forma podremos ejecutar el f-prot con las rutas comunes, y las ligas harán que el ejecutable esté disponible.

```
ln -fs /usr/local/f-prot/f-prot.sh /usr/local/bin/f-prot
chmod +x /usr/local/f-prot/f-prot*
```

Podemos hacer una prueba llamando a la ayuda.

```
f-prot -help
```

Con esto quedó instalado el antivirus adecuadamente. Como en el caso del sistema operativo, el tener las definiciones de los virus es importante. La definición de virus es un archivo binario que contiene las cadenas que identifica cuando algún archivo tiene virus. Los virus son código malicioso que regularmente vienen escondidos por algún archivo. En medio del código del archivo viene el código del virus. La máquina antivirus debe de buscar estas cadenas en la información del archivo. Estas definiciones se actualizan constantemente, tan rápido como la velocidad con que nacen los virus. Actualmente existen cerca de 60000 virus. A continuación muestro las líneas de un pequeño *script* que se conecta al sitio donde se encuentran las definiciones. Este *script* puede ser incluido como parte de un *cron* para que se ejecute cada cierto tiempo. En nuestro caso se ejecuta cada seis horas, por lo que está constantemente verificando por actualizaciones de las definiciones del virus o por alguna actualización de la máquina antivirus. Este es el *script* de actualización:

```
wget -passive-ftp ftp://ftp.f-prot.com/pub/fp-def.zip
wget -passive-ftp ftp://ftp.f-prot.com/pub/macrdef2.zip
unzip fp-def.zip
unzip macrdef2.zip
mv -f SIGN.DEF /usr/local/f-prot/
mv -f SIGN.ASC /usr/local/f-prot/
mv -f SIGN2.DEF /usr/local/f-prot/
```

mv -f SIGN2.ASC /usr/local/f-prot/
mv -f MACRO.DEF /usr/local/f-prot/
*rm -f fp-def.zip**
*rm -f macrdef2.zip**

4.2.5. Detector de virus en correos

Mailscanner es una herramienta de software libre que se encarga de detener los correos, abrirlos (en su caso descompactarlos) y buscar por virus con ayuda de una máquina de identificación de virus. La instalación de *mailscanner* necesita algunas herramientas para poder abrir todos los archivos y buscar por los virus dentro de los correos.

Lo primero que haremos será obtener el *mailscanner* y ponerlo bajo un directorio en el que podamos trabajar en las configuraciones. Éste se obtiene con los siguientes comandos.

```
mkdir /home/javo/Antivirus/Mailscanner
cd /home/javo/Antivirus/Mailscanner/
wget -passive-ftp "http://www.sng.ecs.soton.ac.uk/\
mailscanner/files/4/tar/MailScanner-3.13-2.tar"
```

Lo descompactamos dentro del directorio correcto para las configuraciones

```
cd /usr/local
tar -xvf /home/javo/Antivirus/Mailscanner/\
MailScanner-3.13-2.tar
```

Tenemos que crear algunas configuraciones especiales para el *sendmail*. Este sistema interceptará los correos antes de que se pongan en la cola general de correos. Para esto crearemos algunos directorios.

```
cd /var/spool
mkdir mqueue.in
chown root mqueue.in
chgrp mail mqueue.in
chmod u=rwx,g=rx,o-rwx mqueue.in
```

Trabajaremos sobre el *script* de inicio de *sendmail*, lo respaldaremos y haremos algunas modificaciones en él.

cp /etc/rc.d/init.d/sendmail /etc/rc.d/init.d/sendmail.bak
vi /etc/rc.d/init.d/sendmail

Buscamos la siguiente cadena

```
"daemon /usr/sbin/sendmail $([ "$DAEMON" = yes ] && echo -bd)
```

```
$([ -n "$QUEUE" ] && echo -q$QUEUE)"
```

Y se reemplaza por:

```
sendmail -bd -OPrivacyOptions=noetrn -ODeliveryMode=queueonly \
-OQueueDirectory=/var/spool/mqueue.in
sendmail -q15m
```

Perl es un lenguaje de programación muy poderoso, es un gran generador de reportes y entre sus características principales está el manejo de expresiones regulares. Perl es una gran parte de este software, y necesita algunos módulos para poder encontrar los virus que estamos buscando. A continuación instalaremos estos módulos.

```
mkdir /home/javo/Antivirus/Perl
cd /home/javo/Antivirus/Perl
Instalamos los módulos de perl.
wget "http://www.cpan.org/authors/id/ERYQ/\
IO-stringy-2.108.tar.gz"
cd /usr/src/
tar -zxvf /home/javo/Antivirus/Perl/IO-stringy-2.108.tar.gz
cd /usr/src/IO-stringy-2.108/
perl Makefile.PL
make
make test
make install
cd /home/javo/Antivirus/Perl/
wget "http://www.cpan.org/authors/id/GAAS/\
MIME-Base64-2.12.tar.gz"
cd /usr/src
tar -zxvf /home/javo/Antivirus/Perl/MIME-Base64-2.12.tar.gz
cd /usr/src/MIME-Base64-2.12/
perl Makefile.PL
make
make test
make install
cd /home/javo/Antivirus/Perl/
wget "http://www.cpan.org/authors/id/M/MA/MARKOV/\
MailTools1.43.tar.gz"
cd /usr/src
tar -zxvf /home/javo/Antivirus/Perl/MailTools-1.43.tar.gz
cd /usr/src/MailTools-1.43/
```

```
perl Makefile.PL
make
make test
make install
cd /home/javo/Antivirus/Perl/
wget "http://www.cpan.org/authors/id/R/RB/RBS/\
File-Spec-0.82.tar.gz"
cd /usr/src
tar -zxvf /home/javo/Antivirus/Perl/File-Spec-0.82.tar.gz
cd /usr/src/File-Spec-0.82/
perl Makefile.PL
make
make test
make install
cd /home/javo/Antivirus/Perl/
wget "http://www.cpan.org/authors/id/ERYQ/\
MIME-tools-5.411a.tar.gz"
cd /usr/src
tar -zxvf /home/javo/Antivirus/Perl/MIME-tools-5.411a.tar.gz
cd /usr/src/MIME-tools-5.411/
perl Makefile.PL
make
make test
make install
# Se instala el decodificador TNEF, que se encarga de transcribir

#los ficheros en formato RTF
cd /usr/src/
tar -zxvf /usr/local/MailScanner-3.13-2/mailscanner/bin/\
tnef-1.1.1+sizelimit.tar.gz
ln -sf tnef-1.1.1+sizelimit tnef-1.1
cd tnef-1.1
./configure
make
mv /usr/local/MailScanner-3.13-2/mailscanner/bin/tnef \
/usr/local/MailScanner-3.132/mailscanner/bin/tnef.bak
cp /usr/src/tnef-1.1/src/tnef \
/usr/local/MailScanner-3.13-2/mailscanner/bin/tnef
```

Ahora nos toca configurar el mailscanner. Tendremos que crear algunas ligas y mover algunos archivos de configuración.

```
cd /usr/local/MailScanner-3.13-2/mailscanner/etc/
rm -f mailscanner.conf
ln -fs mailscanner.conf.linux mailscanner.conf
cd /usr/local/MailScanner-3.13-2/mailscanner/
```

Tenemos que buscar la cadena "/opt/mailscanner/" y reemplazarla por "/usr/local/MailScanner-3.13-2/mailscanner/" en los siguientes archivos:líneas.

```
bin/check_mailscanner:virusdir=/opt/mailscanner/bin
bin/check_mailscanner:config=/opt/mailscanner/etc/mailscanner.conf
bin/config.pl:my $prefix = '/opt/mailscanner';
```

Ahora tenemos que reemplazar la línea:

```
etc/mailscanner.conf:Virus Scanner = sophos
```

por:

```
etc/mailscanner.conf:Virus Scanner = f-prot
```

y

```
Sweep = /usr/local/Sophos/bin/sophoswrapper
```

por

```
Sweep = /usr/local/MailScanner-3.13-2/f-prot/f-protwrapper
```

Crearemos algunos links a programas del sistema, puesto que la configuración del *mailscanner* los buscará bajo estas rutas.

```
ln -sf /bin/ps /usr/bin/ps
ln -sf /bin/fgrep /usr/bin/fgrep
ln -sf /bin/grep /usr/bin/grep
ln -sf /bin/sed /usr/bin/sed
```

Y los directorios donde *mailscanner* guardará temporalmente los correos para revisarlos y los directorios donde pondrá los correos con virus.

```
mkdir /var/spool/MailScanner/
mkdir /var/spool/MailScanner/quarantine/
mkdir /var/spool/MailScanner/incoming/
```

Ahora tendremos que crear algunos otros *links* hacias los archivos que necesita el *mailscanner*

```
ln -sf /usr/local/MailScanner-3.13-2/mailscanner \
/usr/local/MailScanner
ln -sf /usr/local/MailScanner-3.13-2/mailscanner/etc/\
filename.rules.conf \
/usr/local/MailScanner-3.13-2/mailscanner/etc/filename.rules
ln -sf /usr/local/MailScanner-3.13-2/mailscanner/etc/\
```

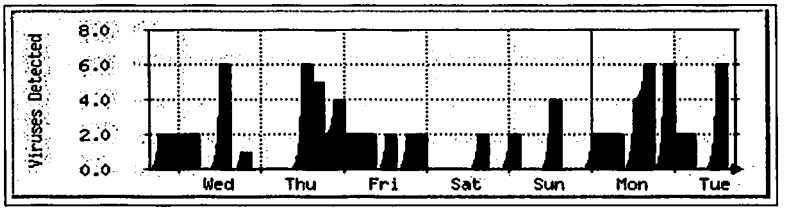


Figura 4.3: Virus recibidos por día

```
sender.virus.report.txt \
/usr/local/MailScanner-3.13-2/mailscanner/etc/\
sender.report.txt
```

Y ponemos un *cron* que compruebe que el *mailscanner* siempre esté funcionando y listo para captar los correos con virus. El *cron* debe de contener las siguientes líneas:

```
# 18/05/2000 JKF Ensure my e-mail virus scanner is still
#running
0,20,40 * * * * [ -x /usr/local/MailScanner-3.13-2/\
mailscanner/bin/check_mailscanner \ ]
 && /usr/local/MailScanner-3.13-2/mailscanner/\
bin/check_mailscanner>/dev/null 2>&1
```

Por último reiniciamos los demonios de *sendmail* para que tome los cambios

```
/etc/rc.d/init.d/sendmail restart
```

Con esto tendremos funcionando nuestro antivirus para los correos.

Esta herramienta nos puede proveer de gráficas que nos muestren cómo se comporta nuestro servidor de correo. Esto se realiza con una herramienta llamada MRTG. Las gráficas nos muestran los virus que hemos recibido por día (Figura 4.3), por semana (figura 4.4) e incluso por año. También podemos obtener información como la cantidad de correos que recibimos, como se ve en la figura 4.4.

Para hacer esto se utilizó un pequeño script que muestro en el anexo A.

Referencias

http://www.sng.ecs.soton.ac.uk/mailscanner/
http://webmaster.bankhacker.com/mailscanner/

TESIS CON
FALLA DE ORIGEN

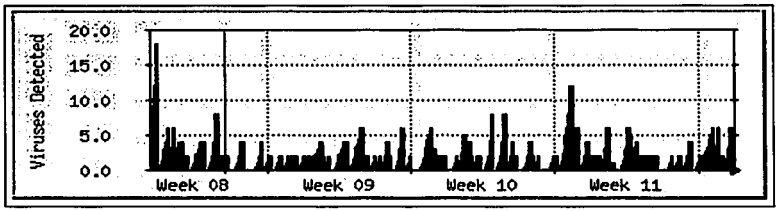


Figura 4.4: Virus recibidos por semana

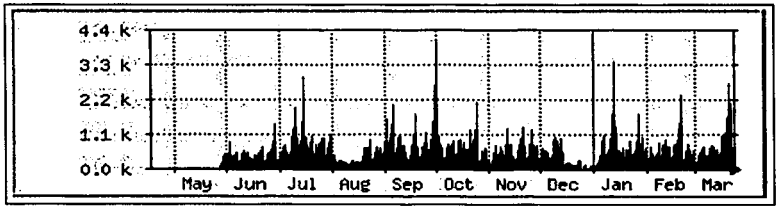


Figura 4.5: Correos recibidos por mes

TESIS CON
FALLA DE ORIGEN

http://www.sng.ecs.soton.ac.uk/mailscanner/mrtg.shtml
http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/
http:/fisica.unam.mx/mrtg/viruses/viruses.html
http:/fisica.unam.mx/mrtg/mail/mail.html

Conclusiones

La seguridad informática es una gran tarea, día a día se descubren nuevos problemas de seguridad y es necesario que todos tomemos conciencia de este gran problema.

Internet y las redes son una gran herramienta, nos vemos envueltos con estos elementos de la informática todos los días, pero para poder sacar una mayor ventaja de ellos debemos de conocer más a fondo sus limitaciones.

En esta tesis trato de exponer algunos de los puntos más vulnerables que podemos encontrar en los servicios de *Internet*, así como exponer algunas posibles soluciones.

El administrador de un servidor, de una red o de un conjunto de máquinas que están conectadas a la *Internet* tiene una gran tarea, y es una tarea que se mueve constantemente, por lo que las personas encargadas de esto siempre deben de estar al día en las noticias, visitar sitios de *hackers* e incluso tratar de probar su seguridad, haciendo escaneos e intentos de penetración a su propia red y servidores. Pero éstas son tareas que pueden estar prohibidas por alguna autoridad, por lo que es necesario siempre tener los permisos necesarios y el respaldo de nuestros superiores ya que no todos los ataques informáticos provienen del exterior, muchas veces son los mismos empleados de la empresa los que quieren sacar algún beneficio personal al obtener información confidencial para tratar de lucrar con ella. Los mecanismos de seguridad deben de ser tanto internos como externos para evitar cualquier tipo de problema.

Como se pudo ver en el caso específico del IFUNAM, una red puede funcionar sin ninguna preocupación sobre la seguridad, y en realidad en la mayoría de las redes de la UNAM sucede esto, pero es sumamente importante el tomar conciencia de que alguien puede estar utilizando nuestra red para atacar a alguien más, o puede estar viendo nuestra información o tratando de robar algún artículo valioso que tenemos en nuestra máquina. Es por esto

que es sumamente importante el tomar conciencia y tratar de hacer algo, tal vez pensemos que sólo las empresas grandes o aquellas que tienen algo que perder pueden ser objetivo de un ataque, y no es así, cualquier máquina conectada a una red puede ser un objetivo de ataque, ya sea por diversión o por obtener algún tipo de beneficio de esa máquina o red.

Existe una gran cantidad de ataques, pero de la misma forma existe una gran cantidad de cosas que podemos hacer para poder prevenirnos, e incluso para poder tomar acciones en el caso de que nos estén atacando. Como lo vimos en esta tesis, existe una gran cantidad de estrategias de seguridad, pero lo mejor que se puede hacer es tomar lo mejor de cada una de ellas. La mejor forma de protegernos será aquella con la cual nosotros podamos ver que tenemos un mejor nivel de seguridad. Existe una gran cantidad de *sites*, libros, revistas y tesis que tratan de ayudarnos en esta tarea, pero si a pesar de esto se decide tomar una posición totalmente pasiva, estamos tomando un camino hacia la indiferencia del mal que podemos hacer ya que nuestra red y máquinas pueden ser utilizadas para atacar a algún otro lugar, y sobre todo estamos dejando nuestra información y nuestros recursos totalmente disponibles, ya sea para explotarlos o simplemente para publicarlos.

Muchas veces las herramientas no son fáciles de instalar, y sobre todo en los sistemas Unix porque en la mayoría de las ocasiones necesitas bajar los fuentes y compilarlos, pero esto dará la seguridad de lo que se esté instalando, y en caso de un administrador conciente de sus tareas, esto es sumamente importante. Aquí se mostraron algunas instalaciones y pudimos observar que en muchas ocasiones es algo tedioso, pero es necesario si queremos tener un buen nivel de seguridad. El instalar las herramientas nos puede llevar mucho tiempo la primera vez, pero debemos de escribir bitácoras de nuestras instalaciones para futuras consultas, e incluso para ayudar a alguien más si lo necesitara.

Algo importante en este rubro es el observar que todo se hizo con software libre, no se tuvo que pagar dinero por una licencia o se tuvo que hacer alguna petición especial para poder utilizarlo. Con esto comprobamos una vez más que el software libre es una alternativa igual o de mayor importancia que el software propietario, pues además de darnos la estabilidad característica de este tipo de software, nos da la posiblidad de instalar una gran cantidad de herramientas bajo la misma licencia que nos ayudará a reforzar todo nuestro sistema de seguridad.

La seguridad no sólo tiene que ver con la privacidad de nuestra información, también tiene que ver con el respeto que podemos tener a otras redes,

o incluso a las personas con las que trabajamos ya que si nuestra máquina tiene virus puede infectar a todas las máquinas que están en nuestra red de trabajo, o si yo acostumbro prestar mi cuenta, la persona a la que se la prestó puede sacar algún tipo de provecho y robar información de la empresa o de los otros usuarios.

Anexo A

En este anexo se muestra el *script* que se ejemplifica en el capÃtulo cuatro para realizar las grÃ¡ficas de MRTG para los virus y *spam*

```
#!/usr/bin/perl
#
# JKF 3/1/2001 Analyse the sendmail logs from magpie and
#crow to find
# the number of viruses rejected and pieces of spam
# detected in a day.
$cmd = shift;
#system("echo >>/tmp/foo.log");
#system("date >>/tmp/foo.log");
#system("/usr/bin/echo cmd is $cmd >>/tmp/foo.log");
$LogDir = "/opt/mrtg/var/sendmaillogs";
#$LogDir = "/var/log";
chdir $LogDir or die "Cannot change to log directory \
$LogDir, $!";
# Find all the compressed files in the log directory
do {
opendir(LOGDIR, '.') || die "Cannot access log dir $LogDir, $!";
while ($file=readdir LOGDIR) {
next if $file =~ /^\./;
push @logfiles, $file if -f $file;
}
sleep 300 unless @logfiles;
} until @logfiles;
# Reset counters
$TotalMails = 0;
$TotalViruses = 0;
```

```
$TotalSpam = 0;
#system("/usr/bin/echo logfiles are " . \
#join(',', @logfiles) .  "
# >>/tmp/foo.log");
# Work through each file building stats
foreach $file (@logfiles) {
if ($file =~ /\.Z$/) {
open(LOG, "zcat $file|")
or (warn("Cannot access log file $file, skipping, $!"), next);
} else {
open(LOG, $file)
or (warn("Cannot access log file $file, skipping, $!"), next);
}
while(<LOG>) {
chomp;
if (/sendmail/) {
$TotalMails += $1 if /nrcpts=(\d+),/;
next;
}
if (/mailscanner/) {
$TotalViruses += $1 if /found (\d+) viruses in/i;
$TotalSpam++ if /message [^\s]+ is spam/i;
}
}
close LOG;
}
print "$TotalMails\n" if $cmd =~ /mail/i;
print "$TotalViruses\n" if $cmd =~ /virus/i;
print "$TotalSpam\n" if $cmd =~ /spam/i;
print "0\n";
print "Not Applicable\n";
print "ECS Mail Servers\n";
```

Bibliografía

[1] Elizabeth D. Zwicky. "Building Internet Firewalls (2nd Edition)". Ed. O´Reilly. USA, 2000.

[2] Wes Sonnenreich. "Building Linux and OpenBSD Firewalls". Ed. John Wiley & Sons, Inc.

[3] Gallo Michael and Hancock William. "Comunicacion entre computadoras y tecnologias de redes". Ed. Thomson Learning. Mexico, 2002.

[4] Kirch Olaf & Dawson Terry. "Linux Network Administrator's guide". Ed. O'Reilly. USA, 2000.

[5] Nemeth Evi & Snyder Garth. "Unix System Administration Handbook (3rd edition)". Ed. Prentice Hall PTR. USA, 2001

[6] Viega John & McGraw Gary. "Building Secure Software". Ed. Addison Wesley. USA, 2002.

[7] Wall Larry & Christiansen Tom. "Programming Perl (3rd edition)". Ed. O'Reilly. USA, 2002.

[8] Blanck-Edelman David. "Perl for System Administration (5st edition)". Ed. O'Reilly. USA, 2000

[9] Stevens Richard. "TCP/IP Illustrated Volume 1". Ed. Addison Wesley. USA, 1994.

[10] Wright Gary, Stevens Richard. "TCP/IP Illustrated Volume 2". Ed. Addison Wesley. USA, 1994.

[11] McClure Stuart, Scambray Joel. "Hacking exposed". Ed. Osborne. USA, 1999.

Ligas con información sobre *Firewalls*

[12] http://www.openbsd.org/

[13] http://www.openbsd.org/faq/faq4.html

[14] http://www.openbsd.org/faq/faq6.html

[15] http://www.openbsd.org/cgi-bin/man.cgi

[16] http://www.openbsd.org.mx/~alex/openbsd_despues_de_la_inst/index.html

[17] http://www.openbsd.org.mx/~alex/conf_fw_openbsd/index.html

[18] http://www.openbsd.org.mx/~alex/SS_OpenBSD/index.html

[19] http://online.securityfocus.com/infocus/1182

[20] http://www.jmu.edu/computing/info-security/engineering/proj/fw/personal.sh

[21] http://www.deadly.org/pf-howto/html/

Ligas con información sobre Seguridad

[22] http://www.cert.org/

[23] http://www.gwolf.cx/seguridad/recom_seg_linux/

[24] http://www.cert.org/current/

[25] http://www.cert.org/current/scanning.html

[26] http://www.securityfocus.com/

[27] http://www.nwfusion.com/topics/dos.html

[28] http://www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html

[29] http://www.faqs.org/faqs/computer-virus/faq/

Ligas con información sobre servidor de correos

[30] http://www.sng.ecs.soton.ac.uk/mailscanner/

[31] http://webmaster.bankhacker.com/mailscanner/

[32] http://www.sng.ecs.soton.ac.uk/mailscanner/mrtg.shtml

[33] http:/fisica.unam.mx/mrtg/viruses/viruses.html

[34] http:/fisica.unam.mx/mrtg/mail/mail.html

Ligas con información sobre Unix

[35] http://www.tldp.org/

[36] http://www.redhat.com/docs/manuals/linux/RHL-7.2-Manual/

[37] http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/

Ligas con información sobre servidor Web

[38] http://www.redhat.com/support/resources/faqs/RH-apache-FAQ/book1.html

[39] http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html_single/SSL-RedHat-HOWTO.html

[40] http://www.redhat.com/support/resources/faqs/RH-apache-FAQ/c163.html#AEN165