UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGÓN

"PROPUESTA PARA LA IMPLEMENTACIÓN DE UN FIREWALL EN EL DEPARTAMENTO DE ADMINISTRACIÓN DE SERVIDORES, DGSCA."

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A:

ERIKA HERNÁNDEZ VALVERDE

ASESOR DE TESIS:
ING. RODOLFO VÁZQUEZ MORALES

MÉXICO, 2004.

"El secreto de la felicidad no está en hacer lo que se quiere, sino en querer siempre lo que se hace". León Tolstoi

DEDICATORIAS

A las dos madres que tengo, Socorro y Gloria, por todo su amor y fe depositados en mi, un especial agradecimiento a ti mama Socorro por ser una inspiración de fortaleza.

A Nelly y Maria Elena, por su apoyo y cariño en todo momento, gracias por confiar en mi.

A mis hermanos y primos, como un estímulo de superación.

A mi familia, a todos los que de una u otra manera conforman mi familia, gracias por sus palabras de aliento.

A mis amigos, a todos los que han pasado por mi vida y en su momento me apoyaron, los que aun permanecen gracias por confiar, apoyar y darme la oportunidad de crecer a su lado, igualmente, agradezco su cariño.... todos son bien correspondidos.

AGRADECIMIENTOS

Escribir los agradecimientos hacia todas las personas que hicieron posible la realización del presente trabajo no es nada simple, probablemente porque a veces las palabras no alcanzan para expresar los sentimientos.

Deseo agradecer de una forma muy especial al Ing. Rodolfo Vázquez Morales, por haber aceptado dirigirme en este trabajo, por su enorme colaboración y apoyo y por sus valiosas y acertadas observaciones que enriquecieron el presente trabajo.

Deseo agradecer a los ingenieros José Manuel Quintero Cervantes, Ricardo Gutiérrez Orozco, Octavio Francisco Mejía Sandoval y Esteban Ayala Peña por sus valiosas y acertadas observaciones que enriquecieron el presente trabajo.

A si mismo a todas las personas que ayudaron en la revisión para la conformación del presente trabajo enriqueciéndola con sus acertadas observaciones.

AGRADECIMIENTOS DEL ASESOR

Universidad Nacional Autónoma de México:
Gracias por brindarme un espacio en tus aulas universitarias y seguir siendo mi casa.

Erika:
La primera vez que impartí una clase tuve miedo y nervios, pero cuando salí del salón estaba contento, era algo nuevo que había disfrutado, hoy, día con día recuerdo ese momento y confirmo que la oportunidad que me dio la vida de poder compartir lo que sé es maravilloso. Gracias por confiar en mi y demostrarme que siempre hay algo nuevo por hacer, eso me compromete, nunca lo voy a olvidar.

A nuestros revisores:

Ing. Ricardo Gutiérrez Orozco:
Gracias por seguir siendo mi maestro y ayudarme a aprender cosas nuevas, con mucho cariño y respeto.

Ing. Esteban Ayala Peña:
Eres un gran ejemplo de entereza, te agradezco el haber revisado este trabajo.

Ing. Octavio Mejía Sandoval:
Gracias por compartir tu entusiasmo hacia la cátedra y por las aportaciones a este trabajo.

Ing. José Manuel Quintero Cervantes:
Gracias por el tiempo y recomendaciones que dedicaste a este trabajo, eres un buen amigo.

Ing. Gladis Fuentes Chávez:
Gracias por ser la piedra angular de mi vida.

Ing. Rodolfo Vázquez Morales.

INTRODUCCIÓN

En la actualidad las razones de la popularidad y el éxito de Internet se deben al hecho de que es una red abierta, esto permite que cualquier red y cualquier máquina puedan conectarse fácilmente. La facilidad de acceso y popularidad son los principales atractivos para las organizaciones que desean obtener las grandes ventajas de este servicio, pero también es la causa de que Internet sea blanco de personas con propósitos inadecuados.

En realidad cualquier calle o casa corre el riesgo de que personas indeseables acceda a ellas. Cualquier transferencia de datos en Internet corre riesgo también, realizar actividades delictivas a través de Internet requiere conocimientos técnicos sofisticados, pero esto no detiene a los intrusos, sino los alenta.

Por lo tanto, la seguridad resulta cada vez más importante, ya que Internet proporciona una poderosa herramienta para distribuir información entre las organizaciones y obtener información de otros, pero no del todo seguro. La forma más segura de evitar daños a la información de la organización es impedir que las personas no autorizadas puedan acceder a las computadoras.

Este trabajo no aborda el extenso panorama de la seguridad en cómputo, sólo se enfoca en uno de los componentes de ésta, que es el Firewall y lo útil que puede ser cuando se implementa en conjunto con otras herramientas de seguridad, las cuales se describirán brevemente, el objetivo principal es la elección de un Firewall y las pruebas del mismo para el Departamento de Administración de Servidores en la Dirección General de Servicios de Cómputo Académico (DGSCA) de la Universidad Nacional Autónoma de México (UNAM).

En el capítulo 1 se muestra un panorama general de Internet, así como algunos de los servicios del mismo, además, por medio de un análisis se muestra lo importante que es saber los recursos con los que cuenta la organización.

En el capítulo 2 se habla de algunas generalidades de Firewalls, arquitectura de los mismos y filtrado de paquetes.

En el capítulo 3 se enfocará a la configuración, implementación y monitoreo del Firewall.

En el capítulo 4 se hablará sobre políticas de seguridad y como se realizan las mismas.

En el apéndice A se describe Iptables.

En el apéndice B se muestran herramientas de seguridad y algunos servicios de red UNIX.

En el apéndice C se muestran políticas y reglas del Firewall. Políticas del Departamento.

Este trabajo pretende dar un panorama de la importancia de la implementación de un mecanismo de seguridad, esperando despertar la inquietud para buscar otras herramientas que configuradas correctamente trabajen en conjunto para proteger la red de cualquier organización.

CAPÍTULO

1

¿PARA QUÉ IMPLEMENTAR UN FIREWALL?

1 ¿Para qué implementar un Firewall?

Se tratará de explicar algunos de los motivos que nos hacen reflexionar si es o no necesario colocar algún mecanismo de protección en el Departamento de Administración de Servidores de la Dirección General de Servicios de Cómputo Académico (DGSCA), lugar en el que se implementará nuestro análisis o en cualquier otro sitio, en donde se tengan equipos que brindan servicios a través de la red. Por lo que se hablará de Internet y de algunos servicios que nos proporciona. Además se dará un panorama que permita analizar los factores a considerar cuando se desea proteger un sitio con cualquier mecanismo de seguridad.

1.1 Panorama general de la seguridad en Internet

Internet en la actualidad

En la actualidad las organizaciones que van a incorporase o las que ya están incorporadas a Internet se encuentran con la preocupación de la seguridad que tiene su información a través de la red, sin embargo, a pesar de la inseguridad que existe, saben que Internet es un medio necesario en nuestros días, ya que los avances que tiene son muy extensos, por ejemplo brindar acceso a una gran cantidad de información o la facilidad para publicarla, hacen que Internet además de ser un recurso importante sea también un blanco atractivo para ataques que ocasionan que Internet sea peligroso debido a que puede contaminar, alterar o destruir la información de los sitios. Lo anterior nos habla de la inseguridad de Internet, por lo que la única manera de transitar por él es imponiendo mecanismos de seguridad.

A pesar de saber que Internet es inseguro las organizaciones la utilizan, se debe a que Internet interconecta a instituciones gubernamentales, académicas, comerciales,

etc., a nivel mundial, muchas redes operadas por una multitud de organizaciones están interconectadas para conformarla denominándola "red de redes", permite comunicarse, compartir recursos y datos con personas ubicadas a un lado nuestro o al otro lado del planeta. Una de sus mayores ventajas es que permite tener acceso a enormes cantidades de información en todo el mundo, constituyendo un herramienta de investigación, una puerta comercial, un medio por el cual puede comunicarse con personas de todo el mundo. Para la comunidad científica, Internet es una herramienta esencial e indispensable para la investigación, para los líderes de la industria y el comercio también representa un gran medio de comunicación y poder adquisitivo.

1.2 La seguridad en Internet

El origen de Internet se debió a un plan militar que nace de un proyecto para solucionar la necesidad de intercambiar información de grado no-militar de los centros de investigación científica que trabajaban para la defensa de los Estados Unidos (universidades, laboratorios, etc.). Quienes lo llevaron a cabo eran personas con formación académica, que tenían como premisa fundamental desarrollar un sistema que permitiera interconectar todas las computadoras de esos centros de investigación, cualquiera que fuere su tipo. Así comenzó a desarrollarse lo que hoy conocemos como protocolo TCP/IP, una de las máximas expresiones de la operatividad computacional, pues permite comunicar cualquier tipo de computadoras entre sí.

Una de las leyes fundamentales de la seguridad informática dice que "el grado de seguridad de un sistema es inversamente proporcional a la operatividad del mismo[1]". Esto se refiere a que darle a un sistema un determinado grado de seguridad, aunque sea mínimo, implica imponer algún tipo de restricción, lo que forzosamente disminuirá la operatividad con respecto al estado anterior en el que no se tenía seguridad.

[1] UPM, "Seguridad en el comercio electrónico", http://criptored.upm.es, consultada en 2003.

Por lo tanto, Internet es una red insegura, porque fue diseñada con un alto nivel de operatividad. No está mal que sea insegura, ni se trata de un error de diseño, sino que para que cumpliera la función para la cual se la creó debía tener el más alto grado de operatividad, lo que trae como consecuencia un alto nivel de inseguridad.

Por lo que se ha dicho hasta aquí se pensaría que si Internet es insegura, ¿porque usarla entonces?, bueno pondremos un ejemplo para dar un motivo a utilizarla, además de ser indispensable para comunicarnos con el mundo exterior.

Alguien podría preguntar ¿es seguro transportar una cantidad de dinero considerable por las calle sin tomar ninguna precaución? (y nos referimos a no tomar ningún tipo de prevención, algo así como llevarlos en una bolsa de plástico transparente), a lo que muchos contestaríamos que no es seguro, pero aun así debemos transportar el dinero, entonces si el transporte se hace en un camión blindado la respuesta será probablemente, un sí.

El punto más importante de este razonamiento es que aunque el dinero llegue sano y salvo a su destino en el camión blindado, esto no implica que las calles se hayan vuelto seguras, simplemente se ha encontrado un mecanismo mediante el cual se puede transportar dinero con un grado de seguridad razonable. Este es un concepto fundamental cuando se aplica a Internet (o a cualquier sistema informático) pues indica claramente que, a lo que se le puede dar un grado de seguridad mediante un determinado mecanismo, es a una operación específica, y este mecanismo se debe repetir cada vez que se lleve a cabo una operación similar.

Esta idea es muy diferente a la de creer que, por implantar determinados mecanismos de seguridad automáticos, Internet se vuelve segura. Por lo tanto, si alguien: dice que por determinado medio Internet se vuelve segura, tal vez sepa mucho de algunas cosas, pero nada de seguridad informática. Y el parámetro fundamental a tener en cuenta en este punto, ya sea uno un usuario final o una corporación, es el siguiente, cuando se conectan dos sistemas, uno seguro y otro inseguro, el grado de

seguridad no se promedia, sino que pasa a ser el del más inseguro para todo el sistema.

Por lo tanto, a partir de la conexión de un sistema seguro (nuestro sistema) con otro inseguro (Internet) se deberá aumentar el grado de seguridad. Dicho de otra manera: cada vez que se agregue algo a un sistema que lo vuelva más abierto (por ejemplo una conexión a Internet) se deberá actualizar la estrategia de seguridad informática del mismo.

Entonces la inseguridad de Internet no es un error de diseño, sólo que no se pensó que crecería tanto, ocasionando ataques a los sistemas que están conectados a Internet por personas con malas intenciones.

1.2.1 Ataques y atacantes más comunes en Internet

Existen muchos tipos de ataques que se han llevado a cabo en sistemas conectados a Internet, derivados son los tipos de atacantes y la forma de atacar, sin embargo, no hay un estándar ni clasificación de ellos, por lo que a continuación se hablará de algunos de tantos que se puede encontrar[2].

Tipos de ataques

Intrusión

Los ataques más comunes a los sistemas son las *intrusiones*; con ellas laos atacantes pueden utilizar sus computadoras. La mayoría de los atacantes quieren utilizar sus computadoras como si fueran usuarios legítimos.

Los intrusos tienen muchas formas de obtener acceso, por ejemplo, utilizan ingeniería social (pueden utilizar el nombre de alguna persona y pedir información a la

[2]BRENT, Chapman, ZWICKY Elizabeth. "*Construya Firewalls para Internet*", O'Reilly, México, *1997.*

cual no están autorizados) o tratan de adivinar cuentas y contraseñas (con combinaciones, utilizando ataques por diccionario con ayuda de programas), se muestra un ejemplo en la figura 1.

Ejemplo:

Paso 1. Un escaneo de puertos.

Paso 2. El atacante explota las debilidades identificadas con el fin de obtener privilegios de administrador en las máquinas públicas o desprotegidas.

Paso 3. Si existen relaciones de confianza, el atacante las explota para entrar a las máquinas internas.

Paso 4. El atacante roba información sensible para utilizarla en los intentos de obtener privilegios importantes de los servicios que ofrece el servidor.

Paso 5. El atacante usa las claves robadas para destruir o robar información confidencial de la organización.

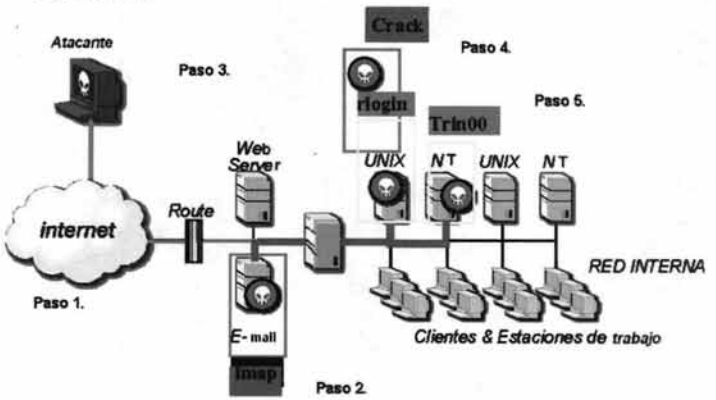


Fig. 1. Ejemplo de un ataque de intrusión.

Negación de servicio

Un ataque de negación de servicio es aquel que está dirigido en su totalidad a evitar que las personas utilicen sus propias computadoras. Aunque algunos casos de sabotaje implican la verdadera destrucción o cierre del equipo y la información, es más frecuente que sigan el patrón de inundación; un intruso inunda a tal grado un sistema o red (con mensajes, procesos o solicitudes a la red) que usted no puede hacer el

verdadero trabajo. El sistema o red pasa todo el tiempo respondiendo a los mensajes o solicitudes y no puede cumplir ninguno de ellos.

Aunque inundar es la forma más simple y común de llevar a cabo a un ataque de negación del servicio, un intruso más hábil, también, puede inhabilitar los servicios, volverlos a enrutar o reemplazarlos.

Es casi imposible evitar todos los ataques de negación de servicio. A veces, para los atacantes es una situación de 'cara', yo gano; 'cruz', usted pierde. Por ejemplo, muchos sitios configuran las cuentas para que no se puedan utilizar después de un cierto número de intentos fallidos para iniciar una sesión, lo cual evita que los atacantes simplemente prueben contraseñas hasta que encuentren la correcta. No obstante, proporciona a los atacantes una forma fácil de montar un ataque de negación del servicio: bloquean la cuenta de cualquier usuario de manera sencilla al intentar varias veces iniciar una sesión.

La mayoría de las veces no se puede evitar el riesgo de ataques de negación del servicio. Si acepta cosas del universo externo (correo electrónico, llamadas telefónicas o paquetes) es posible inundarse. En el mundo electrónico es tan posible que suceda la negación del servicio por accidente como a propósito.

Lo más importante es configurar los servicios de tal modo que si uno se inunda, el resto del sitio continué funcionando mientras se encuentra y se repara el problema. Por fortuna, los ataques intencionales de negación del servicio no son muy populares, son tan fáciles que muchos atacantes los consideran "antideportivos"; tienden ser simples de rastrear y, por lo tanto, son riesgosos para el atacante; y no proporcionan al atacante la información o la habilidad de utilizar las computadoras, la cual es recompensa para la mayoría de los intrusos. Los ataques intencionales de negación del servicio son el trabajo de personas que están enojadas específicamente con un sitio, lo cual no es muy común. Es más probable que se encuentren problemas de negación del servicio no intencionales.

Robo de información

Algunos tipos de ataques permiten que el atacante obtenga información sin tener que utilizar directamente sus computadoras. Por lo general estos atacantes se aprovechan de los servicios de Internet que tienen como fin proporcionar información, haciendo que den más información de lo que era su intención, o dándosela a las personas equivocadas. Muchos servicios de Internet están diseñados para usuarios en redes de área local y no tienen el tipo o grado de seguridad que permitiría de manera segura a través de Internet.

El robo de información no necesita ser activo o especialmente técnico. Las personas que quieran saber información personal pueden llamar y preguntar (tal vez fingiendo ser alguien que tiene derecho a saber), esto es un robo activo de información o pueden intervenir su teléfono, un robo pasivo de información. De manera similar, las personas que quieren reunir información electrónica pueden indagarla activamente (quizá fingiendo ser una máquina o un usuario con acceso válido) o pueden intervenir de manera pasiva la red y esperar que la información fluya.

La mayoría de las personas que roban información intentan tener acceso a sus computadoras; buscan nombres de usuarios y contraseñas.

Intervenir redes es mucho más fácil que intervenir una línea de teléfono. En las tecnologías más comunes de redes, como Ethernet y Token Ring, cualquier computadora que está en una red de área local es capaz de ver todo el tráfico que pasa por la red. El tráfico que cruza Internet puede pasar un sin número de redes de área local, cualquiera de las cuales puede ser un punto de riesgo. Los proveedores de servicio de red y los sistemas de acceso público son blancos muy comunes para intrusiones; analizadores colocados ahí pueden tener mucho éxito porque hay mucho tráfico que pasa a través de estas redes.

Resulta difícil conseguir las estimaciones actuales del número de intentos de intrusión, cualesquiera que sean los números, no hay duda que los intentos globales de piratería en Internet y su nivel de sofisticación están creciendo. Los modelos de las exploraciones de puerto han cambiado desde simples sondeos de unos pocos indicadores de seguridad hasta una exploración de todo el dominio del intervalo completo del puerto del servicio.

Hay organizaciones que se encargan de dar seguimiento a los problemas de ataques, una de ellas es el CERT, el *Computer Emergency Response Team* (CERT), es un equipo informático de respuestas de emergencia, un centro de coordinación de información de seguridad de Internet creado en el Instituto de Ingeniería de Software de la Universidad Carnegie Mellon después del incidente Internet Worm, en 1988. Su sitio principal es www.cert.org de donde se tomaron los siguientes datos que dan un panorama general de lo importante que es implementar seguridad los sistemas.

2000-2003 CERT/CC ESTADÍSTICAS (Enero 22, 2004)

Número de incidentes reportados

Año	2000	2001	2002	2003
Incidentes	21,756	52,658	82,094	137,529

Vulnerabilidades reportadas

Año	2000	2001	2002	2003
Vulnerabilidades	1,090	2,437	4,129	3,784

Alertas de seguridad publicadas en el CERT.org

Año	2000	2001	2002	2003
Consultas	22	37	37	28
Resúmenes	4	4	4	4
Totales	26	41	41	32

Notas de seguridad publicadas

Año	2000	2001	2002	2003
Notas de Incidentes	10	15	6	4
Notas de Vulnerabilidades	47	326	375	255
Notas Totales	57	341	381	259

Mensajes por e-mail manejados

Año	2000	2001	2002	2003
Correos	56,365	118,907	204,841	542,754

Llamadas telefónicas recibidas

Año	2000	2001	2002	2003
Llamadas	1,280+	1,417+	880+	934+

Tipos de atacantes

Todos los atacantes comparten una característica: no quieren ser atrapados, así que intentan ocultarse. Si obtienen acceso a su sistema probablemente intenten conservarlo, por lo que construyen formas adicionales de obtención de acceso (y esperan que dichas rutas no sean descubiertas). La mayoría tiene algún tipo de contacto con otras personas que tienen los mismos tipos de interés ("los subterráneos"), y la mayoría comparte la información que se obtiene de atacar a un sistema.

Joyriders

Los joyriders son personas aburridas que buscan alguna diversión. Entran porque piensan que usted puede tener datos interesantes; porque sería divertido utilizar sus computadoras o porque no tienen nada mejor que hacer. Tal vez quieran saber que tipo de computadora tiene o los datos que posee. Son curiosos, pero no activamente maliciosos; sin embargo, con frecuencia dañan el sistema por ignorancia o por intentar cubrir su rastro. En especial les atrae los sitios bien conocidos y computadoras poco comunes.

Vándalos

Los vándalos quieren causar daño, ya sea porque gozan con destruir cosas o porque usted no les agrada. Los vándalos son un gran problema si usted es alguien que el underground de Internet considera como el enemigo (por ejemplo, la compañía de teléfonos o el gobierno), también puede convertirse en un blanco sólo por ser grande y visible.

Por fortuna los vándalos no son muy comunes, los vándalos obligan a la gente a tomarse muchas molestias para encontrarlos y detenerlos. La mayoría va directo a la destrucción, que es desagradable pero relativamente fácil de detectar y reparar. En la mayoría de las circunstancias, eliminar sus datos o arruinar su equipo no es lo peor que alguien pueda hacerle, pero es lo que hacen los vándalos (en realidad, introducir cambios sutiles pero significativos en los programas o datos financieros es más difícil que detectar y reparar).

Score Keepers

Al igual que los joyriders y vándalos, los score keepers pueden preferir sitios de interés particular. Irrumpir en algo bien conocido, bien defendido o bien ordenado significa puntos más valioso para ellos. Sin embargo, también atacarán a cualquiera que esté a su alcance; persiguen cantidad así como calidad. No necesariamente desean algo que usted tiene, ni les importa en lo más mínimo las características de su sitio. Pueden o no hacer daño a su paso.

Con toda seguridad reunirán información que guardarán para utilizarla posteriormente (tal vez para intercambiarla con otros atacantes). Es probable que intenten dejar formas de volver después y, si es posible, utilizarán sus máquinas como plataforma para atacar otros sitios.

Estas personas se descubren mucho después de haber entrado en su sistema. Quizá usted se de cuenta lentamente, porque algo raro pasa con su máquina.

Espías

La mayoría de las personas que entran sin permiso a las computadoras lo hacen por la misma razón que la gente escala montañas: porque están ahí. Aunque estas personas no están muy interesadas en el hurto, es común que roben cosas que se puedan convertir directamente en dinero o en mayor acceso (tarjetas de crédito, teléfono o información para acceso a redes). Si encuentran secretos que creen que pueden vender, tal vez intenten hacerlo, pero no es su negocio principal.

Como hipótesis se podría mencionar que un Firewall podría evitar o aislar de algunos ataques y atacantes de este tipo, así como de la inseguridad de Internet, pero eso se afirmará sólo cuando se tenga más conocimiento de lo que es un Firewall y de lo que puede librar, por lo que en los siguientes capítulos se verá.

1.3 Servicios de Internet

Al intentar implementar un Firewall para Internet una de las preocupaciones es proteger los servicios que se utilizarán y proporcionarán a través del Internet. Se describirán algunos de los principales servicios de Internet, con lo cual se espera dar un panorama general de ellos y sabes que servicios utilizar en la organización.

Los servicios basados en red son programas que se ejecutan en una máquina a los que pueden acceder otros equipos de la red. Los puertos de servicio identifican los programas y las sesiones individuales o las conexiones que se realizan. Los puertos de

servicio son los nombres numéricos para los diferentes servicios de red, también se usan como identificadores numéricos para los extremos finales de una conexión particular. Los números de puerto de servicio van desde 0 hasta 65535.

Los programas de servidor (por ejemplo, los demonios o *daemons*) escuchan las conexiones entrantes en un puerto de servicio asignado a ellos. Por convención, los servicios de red principales están asignados a números de puerto bien conocidos, en el intervalo inferior, desde 1 a 1023. Estas asignaciones numéricas de puerto a servicio las coordina la autoridad de Asignación de números de Internet (IANA).[3]

Estos puertos de intervalo inferior se llaman puertos privilegiados porque sus propietarios son programas que ejecutan privilegios del nivel del sistema, es decir, súper usuario o *root*.

Los números de puerto superiores, desde 1024 hasta 65535, se llaman puertos no privilegiados. Sirven para un propósito doble. Casi siempre, estos puertos se asignan de forma dinámica al cliente de una conexión. La combinación de pares de números de puerto cliente y servidor, junto con sus respectivas IP, identifican unívocamente la conexión.

En los sistemas UNIX podemos ver una lista de los números de puerto de servicio comunes en el archivo /etc/services.

Cada entrada consta de un nombre simbólico para un servicio, el número de puerto asignado a él, el protocolo (TCP o UDP) sobre el que se ejecuta el servicio y cualquier alias opcional para el servicio. La siguiente tabla muestra algunas asignaciones habituales de nombre de servicio a número de puerto, tomada de la versión 8 de Red Hat.

[3] IANA: Internet Assigned Number Authoriry, http://www.iana.org

```
service-name  port/protocol  [aliases ...]
--------------------------------------------

ftp-data      20/tcp
ftp-data      20/udp
ftp           21/tcp
ftp           21/udp
ssh           22/tcp
ssh           22/udp
telnet        23/tcp
telnet        23/udp
smtp          25/tcp          mail
smtp          25/udp          mail
nameserver    42/tcp          name
nameserver    42/udp          name
nicname       43/tcp          whois
nicname       43/udp          whois
domain        53/tcp          nameserver
domain        53/udp          nameserver
http          80/tcp          www www-http
http          80/udp          www www-http
..
..
```

Tabla1. Muestra parte de los servicios que se encuentran en el archivo /etc/services

Si se piensa que un servicio no es necesario o que el riesgo que representa para el sistema es mayor al beneficio que representa se puede no habilitar o deshabilitar dicho servicio, en el Apéndice B en la tabla A1 se pueden ver detalles de algunos servicios, en particular en el Sistema Operativo UNIX, así como algunas recomendaciones para ellos.

1.3.1 Correo electrónico

El correo electrónico es uno de los servicios de redes más populares y básicos, desde el punto de vista de usuarios esencial. Por desgracia también es uno de los más

vulnerables, los servidores para correo son blancos muy tentadores porque aceptan datos arbitrarios de anfitriones externos arbitrarios.
Falsificar correo electrónico es sencillo (como falsificar correo postal normal), aceptar correo electrónico ocupa tiempo en la computadora y espacio en el disco, exponiéndolo a ataques.

El Protocolo SMTP[4] es el protocolo estándar de Internet para enviar y recibir correo electrónico. SMTP en sí no es un problema de seguridad, pero lo pueden ser los servidores de SMTP. Un programa que entrega correo a usuarios con frecuencia necesita la capacidad de ejecutarse como cualquier usuario que recibe correo. Esto le da poder amplio y lo hace un blanco tentador.

Si le preguntamos a cualquier administrador de máquinas UNIX, con algo de experiencia, ¿cuál ha sido el software que más problemas de seguridad le ha causado?, nos responderá, sin dudarlo: *sendmail*, por supuesto.

No sólo *sendmail* y el protocolo SMTP, sino que también, con la popularización de POP3, los servidores de este protocolo son un peligro potencial de tener en cuenta en cualquier entorno informático donde se utilice el correo electrónico: es decir, en todos.

De entrada, un programa como *sendmail* (lo ponemos como ejemplo por ser el más popular, pero podríamos hablar en los mismos términos de casi cualquier servidor SMTP) proporciona demasiada información a un atacante.

Además, no sólo se proporcionan datos útiles para un pirata como la versión del programa utilizada o la fecha del sistema, sino que se llega incluso más lejos: tal y como se instalan por defecto, muchos servidores informan incluso de la existencia o inexistencia de nombres de usuario y de datos sobre los mismos.

[4]SMTP: Protocolo Simple de Transferencia de Correo(Simple Mail Transfer Protocol)

Independientemente del programa que se utilice como servidor de correo y su versión concreta, con vulnerabilidades conocidas o no, otro gran problema de los sistemas de correo SMTP es el *relay*: la posibilidad de que un atacante interno utilice nuestros servidores para enviar correo electrónico a terceros, no relacionados con nuestra organización. Aunque en principio esto a muchos les pueda parecer un mal menor, no lo es; de entrada, si nuestros servidores permiten el *relay* estamos favoreciendo el *spam* en la red, que se refiere al envío de e-mail no deseado con fines casi siempre publicitarios, algo que evidentemente a nadie le hace gracia recibir.

Además, el *relay* causa una negación de servicio contra nuestros usuarios legítimos, tanto desde un punto de vista estrictamente teórico (alguien consume nuestros recursos de forma no autorizada, degradando así el servicio ofrecido a nuestros usuarios legítimos) como en la práctica: cuando una máquina encuentra un servidor SMTP en el que se permite el *relay* lo utiliza masivamente mientras puede, cargando enormemente la máquina y entorpeciendo el funcionamiento normal de nuestros sistemas. Por si esto fuera poco, si se incluye a nuestra organización en alguna "lista negra" de servidores que facilitan el *spam* se causa un importante daño a nuestra imagen e incluso ciertos dominios pueden llegar a negar todo el correo proveniente de nuestros servidores.

Sólo existe una manera de evitar el *relay*, configurando correctamente todos nuestros servidores de correo. Por supuesto, esto es completamente dependiente de los programas utilizados en nuestro entorno, por lo que es necesario consultar en la documentación correspondiente la forma de habilitar filtros que eviten el *relay*; por Internet existen incluso filtros genéricos para los servidores más extendidos.

1.3.2 Transferencia de datos

FTP (File Transfer Protocol) es el estándar para transferir archivos por Internet. Además, hay algunos protocolos especializados que se usan para aplicaciones en

donde FTP no es apropiado. Dispositivos dedicados emplean TFTP (Trivial File Transfer Protocol) para transmitir archivos de configuración. FSP (File Service Protocol) es un protocolo basado en UDP que se utiliza cuando las conexiones basadas en TCP no funcionan o no están permitidas. UUCP se ocupa para transferir lotes, sobre todo a través de líneas telefónicas.

Protocolo de Transferencia de Archivos (FTP)

FTP se utiliza para transferir archivos de una máquina a otra. Puede utilizarlo para transferir cualquier tipo de archivo, incluyendo binarios ejecutables, de imágenes, texto ASCII, PostScript, archivos de sonido y video entre muchos más. Hay dos tipos de acceso FTP: FTP de usuario y FTP anónimo. El primero requiere de una cuenta en el servidor y permite a los usuarios obtener cualquier archivo como si iniciará una sesión. El segundo es para personas que no tienen una cuenta en el servidor y se utiliza para proporcionar acceso a archivos específicos al mundo en general.

FTP utiliza dos conexiones FTP separadas una para transportar comandos y resultados entre el cliente y el servidor (comúnmente llamada canal de comandos), y otra para transportar cualquier archivo real y listas de directorio transferidas (canal de datos). En el lado del servidor, el canal de comandos utiliza el puerto 21, y el canal de datos normalmente utiliza el puerto 20. El cliente emplea puertos arriba del 1023 para los canales de comandos de datos.

Para iniciar una sesión FTP, un cliente designa primero dos puertos para sí mismo, cada uno de ellos con un número de puerto arriba del 1024. Utiliza el primero para abrir la conexión del canal de comando con el servidor y luego emite el comando PORT de FTP para indicar al servidor el número del segundo puerto, que el cliente quiere utilizar para canal de datos. Entonces el servidor abre la conexión del canal de datos, la cual es contraria a la mayoría de los protocolos, que abren las conexiones del cliente al servidor. Esta apertura inversa complica las cosas para los sitios que intentan hacer filtrado de paquetes al "inicio de la conexión" para asegurar que todas las

conexiones TCP se inicien desde dentro, ya que los servidores FTP externos intentarán iniciar las conexiones de datos con clientes internos, en respuesta a conexiones de comandos abiertas desde tales clientes internos. Además, estas conexiones irán a puertos que se sabe están en un rango inseguro.

1.3..3 Acceso de terminal remota

Los programas que proporcionan acceso de terminal remota permiten que utilice un sistema remoto como si su máquina fuera una terminal conectada directamente.
Telnet es un servicio diseñado para permitir el acceso a un sistema en algún punto de la red. Telnet brinda una "terminal virtual" en la computadora remota.
Telnet se consideró en un tiempo un servicio más o menos seguro porque requería que los usuarios se autentifiquen por ellos mismos. Telnet al enviar sus datos sin cifrar se vuelve vulnerable a ataques de espionaje.

Debido al riesgo del uso de analizadores en la red, usar Telnet para conectarse a un sistema remoto, trae consigo grandes riesgos en cuanto a seguridad. Telnet es seguro sólo si la máquina remota a la cual se conecta y todas las redes que están a su alrededor también son seguras.

Existen programas (además de Telnet), que pueden usarse para tener acceso como terminal remota y ejecución remota de programas (rsh, rlogin), estos programas se utilizan en un ambiente confiable para permitir que los usuarios tengan acceso remoto sin que deban autentificarse nuevamente.
El anfitrión al que se conecta confía en el anfitrión que se está conectando a él; por ende, en los usuarios que provienen de él.

Secure Shell

El servicio Secure Shell (SSH) intenta solucionar problemas se seguridad, destaca el uso de un potente cifrado para los datos transmitidos, con lo que ni las contraseñas ni otros datos pueden ser robados ni siquiera por atacantes que escuchen los datos que se están transmitiendo. El uso de cifrado impide también los ataques en los que el intruso entra en una conexión existente y cambia los datos en las dos direcciones. Los ataques de esta índole pueden usarse para añadir comandos a las sesiones, incluso en los casos en que la sesión haya sido autentificada mediante una segura contraseña de un sólo uso. SSH usa otras técnicas criptográficas para efectuar una potente autentificación de los *hosts* y de los clientes. Esto significa que se puede tener un alto grado de confidencialidad a la que sólo los usuarios autorizados tienen permiso para conectarse.

El proceso de conexión de SSH es el siguiente:

Cuando un cliente se conecta a un servidor SSH, verifica que el servidor sea realmente la máquina a la que se quería conectar. El cliente y el servidor intercambian claves de cifrado (de modo que impide a los espías que se aprendan las claves). El servidor autentifica entonces al cliente, usando el mecanismo de *rhosts*, la autentificación tradicional basada en la contraseña, o bien (de manera más segura) la autenticación RSA. Una vez que el cliente ha sido autentificado, el servidor lanza un *shell* o ejecuta un comando, a petición del cliente.

1.3.4 Worl Wide Web

Hoy en día las conexiones a servidores *web* son sin duda las más extendidas entre usuarios de Internet, hasta el punto de que muchas personas piensan que este servicio (http, puerto 80 TCP) es el único que existe en la red. Lo que en un principio se diseñó para que unos cuantos físicos intercambiaran y consultaran artículos fácilmente, en la actualidad mueve a diario millones de dólares y es uno de los pilares

fundamentales de cualquier empresa: es por tanto un objetivo muy atractivo para cualquier pirata.

WWW es la colección de servidores HTTP en Internet. El *Web* utiliza tecnología de hipertexto para enlazar una gran cantidad de documentos que pueden incluir texto, imágenes, sonido, video y otros formatos. Puede navegar por los documentos de cualquier manera para buscar información. El hipertexto proporciona la posibilidad de navegar de un documento a otro en Internet.

HTTP es el principal protocolo de aplicación que utiliza el *World Wide Web*: proporciona acceso de usuario a los archivos que conforman el servicio *Web*. Por desgracia, los navegadores *Web* y los servidores son difíciles de asegurar. La utilidad del *Web* se basa, en gran medida, en su flexibilidad, pero ésta dificulta su control.

Es necesario garantizar que la información almacenada en la máquina servidora no pueda ser modificada sin autorización, que permanezca disponible y en la que sólo se pueda acceder por aquellos usuarios a los que les este legítimamente permitido.

Cuando un usuario se conecta a un servidor *Web* se produce un intercambio de información entre ambos; es vital garantizar que los datos que recibe el cliente desde el servidor sean los mismos que se están enviando (esto es, que no sufran modificaciones de terceros), y también garantizar que la información que el usuario envía hacia el servidor no sea capturada, destruida o modificada por un atacante. Esto es especialmente importante si la información en tránsito es secreta, como en el caso de los *passwords* que el usuario teclea para autenticarse en el servidor, o en el comercio electrónico, el intercambio de números de tarjetas de crédito.

Por último es necesario garantizar al usuario que lo que descarga de un servidor no va a perjudicar a la seguridad de su equipo; sin llegar a extremos de *applets* maliciosos o programas con virus, si simplemente el navegador del usuario "se cuelga" al acceder al visitar las páginas de una organización, seguramente esa persona dejará

de visitarlas, con la consecuente pérdida de imagen (y posiblemente de un futuro cliente) para esa entidad.

Los problemas relacionados con servidores *web* suelen proceder de errores de programación en los *CGIs* ubicados en el servidor. Un *CGI* (*Common Gateway Interface*) es un código capaz de comunicarse con aplicaciones del servidor, de tal forma que desde una página se invoque a dichas aplicaciones, pasándoles argumentos y el resultado que se muestren en el navegador de un cliente; cuando rellenamos un formulario, vemos una imagen sensible, o simplemente incrementamos el contador de cierta página, estamos utilizando *CGIs*.

1.3.5 Servicios de Información

Muchos usuarios quieren tener acceso a servicios adicionales de información; *Gopher, WAIS y Archie* (en la actualidad no tan populares).
Gopher es una herramienta orientada a menús, basada en texto, que ayuda a los usuarios a encontrar información en Internet. La información de un servidor *Gopher* se organiza como una serie de menús jerárquicos desde los cuales un usuario selecciona elementos. Cada elemento puede ser un archivo, una forma o un menú adicional con sus propios elementos.

En el servicio de Área Amplia (*WAIS* o *Wide Area Information Service*) un usuario hace una consulta sencilla (por lo general una palabra o frase clave) y el servidor *WAIS* devuelve una lista de los documentos que contienen esas palabras junto con un marcador de cada documento. Este marcador se construye con el número de veces que las palabras clave se mencionan y el tamaño del documento.
Archie es un servicio de Internet que busca en los índices se servidores FTP anónimo los nombres de archivos y directorios. Es usual que los servidores *Archie* proporcionen el servicio a través de T*elnet* y correo electrónico, además de los clientes *Archie*.

1.3.6 Servicios de Conferencias en tiempo real

Existen varios servicios para conferencias en tiempo real disponibles en Internet, incluyendo *talk*, IRC y los distintos servicios proporcionados a través de *Multicast Backbone* (MBONE). Todos estos servicios proporcionan un método para que las personas interactúen con otras personas, a diferencia de las bases de datos o archivos de información.
Este servicio requiere que los participantes estén en el momento de la conexión a diferencia del correo electrónico u otros servicios.

1.3.7 Servicio de nombres

El servicio de nombres se encarga de traducir los nombres de anfitrión que utilizan las personas a las direcciones IP numéricas que utilizan las máquinas. Al principio, era posible para cada sitio mantener una tabla de anfitriones con el nombre y número de cada máquina en Internet. Con millones de anfitriones conectados, no resulta práctico para ningún sitio mantener una lista semejante, mucho menos que la tenga cada sitio. En lugar de eso, DNS[5] permite que cada sitio tenga información sobre sus propios anfitriones y pueda encontrar la información para otros sitios. DNS da soporte a SMTP, FTP, Telnet y casi cualquier otro servicio que necesiten los usuarios, quienes quieren escribir "telnet sitio.com" en lugar de "telnet 172.16.0.29".

1.3.8 Servicio de administración de redes

Existe una variedad de servicios utilizados para administrar y mantener las redes, son servicios que la mayoría de los usuarios no emplean porque los desconocen o porque no los necesitan, pero son herramientas muy importantes para los administradores de red.

[5] Servicio de Nombres de Dominio (DNS o Domain Name Service)

Las dos herramientas para administración de redes más comunes son *ping* y *traceroute*. Ambas se conocen como herramientas UNIX, por ser los primeros en utilizarlas, pero ahora están disponibles de alguna forma en casi todas las plataformas en Internet. No tienen sus propios protocolos, ocupan el mismo protocolo fundamental, el protocolo ICMP[6]. A diferencia de muchos de los programas que hemos analizado, no son clientes de algún servidor especifico. ICMP se implementa a bajo nivel como parte indispensable de los protocolos TCP/IP que usan todos los anfitriones en Internet.

El comando *ping* sólo le dice si puede o no hacer llegar un paquete de un punto a otro, información adicional, como cuánto tarda en hacer el viaje de ida y vuelta. *Traceroute* le notifica no sólo si puede llegar a un anfitrión específico sino además la ruta que siguen sus paquetes para llegar a él, lo cual es muy útil para analizar problemas de la red en alguna parte entre los puntos.

El Protocolo SNMP[7] es un protocolo diseñado para facilitar la administración central de equipo de red (enrutadores, puentes, concentradores, computadoras centrales y hasta cierto punto anfitriones). Las estaciones de administración SNMP pueden solicitar información (si una *interface* está arriba o abajo, cuántos *bytes* se han transferido a través de esa *interface*, cuántos errores ha habido en ella, etc.) del equipo de red por medio de SNMP.

1.3.9 Sistemas de archivos de red

Existen varios protocolos disponibles para permitir que las computadoras monten sistemas de archivos que están físicamente conectados a otras computadoras, lo cual es muy deseable porque permite que las personas utilicen archivos remotos sin el gasto que representa transferirlos nuevamente de un lado a otro y tratar de mantener versiones múltiples en sincronía. Es muy peligroso porque significa que permite a las

[6] ICMP: Protocolo de Control de Mensajes de Internet (Internet Control Message Protocol)
[7] SNMP: Protocolo Simple de Administración de Redes (simple Network Management Protocol)

personas leer sus datos sin obtener una autenticación adicional en la máquina donde estos residen. El sistema NFS y el sistema AFS[8] son los dos sistemas para archivos de redes en UNIX que se emplean con más frecuencia. NFS se diseño para usarse en redes de área local y brinda respuestas rápidas, gran confiabilidad, sincronización de hora y un alto grado de confianza entre máquinas. AFS se diseño para usarse a través de redes más grandes; tolera mejor el bajo rendimiento y grados más bajos de confianza.

Existen algunos problemas serios de seguridad con NFS. Si no lo ha configurado de manera adecuada. Un atacante puede sencillamente montarlo en sus sistemas de archivos. En la forma en que funciona NFS, las máquinas cliente tienen permitido leer y cambiar archivos guardados en el servidor sin tener que iniciar una sesión con éste o teclear una contraseña. NFS no registra las transacciones, por lo que no se sabrá si alguien tiene acceso total a sus archivos.

NFS proporciona una forma para controlar qué máquinas pueden tener acceso a los archivos. Un archivo llamado /etc/exports permite especificar qué sistemas de archivos pueden montarse, y qué máquinas pueden hacerlo. Si deja un sistema de archivos fuera de /etc/exports, ninguna máquina puede montarlo. Si lo pone en /etc/exports pero no especifica qué máquinas pueden montarlo, permite que cualquier máquina lo haga.

1.3.10 Sistemas de Ventanas

La mayoría de las máquinas UNIX proporciona en la actualidad sistemas de ventanas basados en X11. El acceso a redes es una característica importante de X11. Aunque cada vez más programas tienen interfaces gráficas de usuario, el acceso de terminal remota se vuelve menos útil: necesita gráficas, no sólo texto. X11 le

[8]NFS: Sistema de Archivos de Red (Network File System), AFS: Sistema de archivos Andrew (Andrew File System)

proporciona gráficas remotas, la desventaja es que lo hace proporcionando acceso total a todas las capacidades que le da cuando se está sentado frente a la máquina.

Los servidores X11 son blancos tentadores para los intrusos. Un intruso con acceso a un servidor X11 puede hacer cualquiera de los siguientes tipos de daño:

Obtener descarga de pantallas:

Copias de cualquier cosa que se muestre en las pantallas del usuario.

Leer las teclas que oprime el usuario

Puede incluir contraseñas de usuario.

Inyectar pulsaciones de teclas

Se verán como si las hubiera oprimido el usuario.

De modo predeterminado, los servidores X11 utilizan autenticación basada en direcciones, si es que emplean alguna; muchos usuarios desactivan está característica por conveniencia. Por lo tanto, X11 no es seguro para usarse a través de Internet. El servidor sí proporciona la opción de utilizar una autenticación más estricta, pero la mayoría de los clientes no son capaces de utilizarla y rara vez está encendida.

1.3.11 Sistemas de impresión

Tanto lp, el sistema de impresión de System V, como lpr, el sistema de impresión de BSD (las dos ramas de UNIX más generales) proporcionan opciones para impresión remota. Estos sistemas permiten que una computadora imprima en una impresora físicamente conectada a otra computadora. Es obvio que en una red de área local esto es muy deseable, pues así no necesita tantas impresoras como computadoras. Sin embargo, las opciones para impresión remota son formas inseguras e ineficientes de transferir datos a través de Internet. No hay razón para permitirlo.

Se listan los servicios en la siguiente tabla y se describe el nivel de uso en el Departamento.

SERVICIO	DESCRIPCIÓN
Correo electrónico	Este tiene un nivel de uso muy alto en el Departamento y en cualquier otro lugar conectado a Internet.
Transferencia de archivos, FTP	En la actualidad es poco utilizado, preferiblemente no se debería usar.
Terminal remota, Telnet, Secure Shell	El acceso a una terminal debería ser sólo por Secure Shell, por las ventajas en cuanto a seguridad se refiere.
WWW	Cualquier sitio conectado a Internet hace uso de este servicio, ya sea en menor o mayor grado.
Servicios de información	Actualmente no se utilizan, y en el Departamento no se utiliza.
Servicios de conferencia	Es utilizado en sitios que se dedican a dar conferencias, en el Departamento no se utiliza.
Servicio de nombres, DNS	Es uno de los servicios más utilizados, y el Departamento no es excepción en su uso.
Servicios de administración de redes	Las dos herramientas que se mencionaron, son de las más utilizadas en cualquier sitio, en el Departamento se utiliza frecuentemente.
Sistema de archivos de red, NFS	Este servicio sólo se debe utilizar en redes de confianza y evitar utilizarlo a través de Internet.
Sistema de ventanas, X11	Este servicio al igual que el anterior, sólo se debe utilizar en redes de confianza.
Sistemas de impresión	Este servicio debe evitarse a través de Internet, en el Departamento de utiliza pero en la red interna.

Tabla 2. Resumen de servicios de Internet

El panorama general de Internet y los servicios de éste, nos hacen comprender el riesgo y la seguridad necesaria para su operación, para lo cual utilizaremos un análisis formal que nos ayude a conocer cuales son los recursos a proteger y también las amenazas de las cuales nos vamos a proteger

1.4 Análisis de riesgos

En un entorno informático existen una serie de recursos (humanos, técnicos, de infraestructura, etc.) que están expuestos a diferentes tipos de riesgos: los "normales", aquellos comunes a cualquier entorno, y los excepcionales, originados por situaciones concretas que afectan o pueden afectar a parte de una organización o a toda la misma, como la inestabilidad política en un país o una región sensible a terremotos. Para tratar de minimizar los efectos de un problema de seguridad se realiza lo que denominamos un **análisis de riesgos**, término que hace referencia al proceso necesario para responder a tres cuestiones básicas sobre nuestra seguridad:

- ¿Qué queremos proteger?
- ¿Contra quién o qué lo queremos proteger?
- ¿Cómo lo queremos proteger?

En la práctica existen dos aproximaciones para responder a estas cuestiones, una cuantitativa y otra cualitativa. La primera de ellas es con diferencia la menos usada, ya que en muchos casos implica cálculos complejos o datos difíciles de estimar. Se basa en dos parámetros fundamentales: la probabilidad de que un suceso ocurra y una estimación del costo o las pérdidas en caso de que así sea.

El segundo método de análisis de riesgos es el cualitativo, de uso muy difundido en la actualidad especialmente entre las "consultoras" de seguridad (aquellas más especializadas en seguridad lógica, cortafuegos, tests de penetración y similares). Es mucho más sencillo e intuitivo que el anterior, ya que ahora no entran en juego probabilidades exactas sino simplemente una estimación de pérdidas potenciales. Para ello se interrelacionan cuatro elementos principales: las amenazas, por definición siempre presentes en cualquier sistema, las vulnerabilidades, que potencian el efecto de las amenazas: el impacto asociado a una amenaza, que indica los daños sobre un activo por la materialización de dicha amenaza: y los controles o salvaguardas, contramedidas para minimizar las vulnerabilidades (controles preventivos) o el impacto

(controles curativos). Por ejemplo, una amenaza sería un pirata que queramos o no (no depende de nosotros) va a tratar de modificar nuestra página *web* principal, el impacto sería una medida del daño que causaría si lo lograra, una vulnerabilidad sería una configuración incorrecta del servidor que ofrece las páginas, y un control la reconfiguración de dicho servidor o el incremento de su nivel de parcheado. Con estos cuatro elementos podemos obtener un indicador cualitativo del nivel de riesgo asociado a un activo determinado dentro de la organización, visto como la probabilidad de que una amenaza se materialice sobre un activo y produzca un determinado impacto.

Tras obtener, mediante cualquier mecanismo, los indicadores de riesgo en nuestra organización llega la hora de evaluarlos para tomar decisiones organizativas acerca de la gestión de nuestra seguridad y sus prioridades. Tenemos por una parte el *riesgo calculado*, resultante de nuestro análisis, y este riesgo calculado se a de comparar con un cierto umbral (*umbral de riesgo*) determinado por la política de seguridad de nuestra organización; el umbral de riesgo puede ser o bien un número o bien una etiqueta de riesgo (por ejemplo, nivel de amenaza alto, impacto alto, vulnerabilidad grave, etc.), y cualquier riesgo calculado superior al umbral ha de implicar una decisión de reducción de riesgo. Si por el contrario el calculado es menor que el umbral, se habla de *riesgo residual*, y el mismo se considera asumible (no hay porqué tomar medidas para reducirlo). El concepto de asumible es diferente al de *riesgo asumido*, que denota aquellos riesgos calculados superiores al umbral pero sobre los que por cualquier razón (política, económica, etc.) se decide no tomar medidas de reducción; evidentemente, siempre hemos de huir de esta situación.

Una vez conocidos y evaluados de cualquier forma los riesgos a los que nos enfrentamos podremos definir las políticas e implementar las soluciones prácticas los mecanismos para minimizar sus efectos. Vamos a intentar de entrar con más detalle en cómo dar respuesta a cada una de las preguntas que nos hemos planteado al principio de este punto:

- Identificación de recursos
- Identificación de amenazas
- Medidas de protección

1.4.1 Identificación de recursos.

Debemos identificar todos los recursos cuya integridad pueda ser amenazada de cualquier forma; por ejemplo, define básicamente los siguientes:

- *Hardware*
  Procesadores, tarjetas, teclados, terminales, estaciones de trabajo, computadoras personales, impresoras, unidades de disco, líneas de comunicación, servidores, *routers*, etc.
- *Software*
  Códigos fuente y objeto, utilidades, programas de diagnóstico, sistemas operativos, programas de comunicación, configuraciones, etc.
- Información
  En ejecución, almacenados en línea, almacenados fuera de línea, en comunicación, bases de datos, etc.
- Personas
  Usuarios, operadores, etc.
- Accesorios
  Papel, cintas, tóners, etc.

Aparte del recurso en sí (algo tangible, como un *router*) hemos de considerar la visión intangible de cada uno de estos recursos (por ejemplo la capacidad para seguir trabajando sin ese *router*). Es difícil generar estos aspectos intangibles de los recursos, ya que es algo que va a depender de cada organización, su funcionamiento, sus seguros, sus normas... No obstante, siempre hemos de tener en cuenta algunos

aspectos comunes: privacidad de los usuarios, imagen pública de la organización, reputación, satisfacción del personal y de los clientes.

Con los recursos correctamente identificados se ha de generar una lista final, que incluirá todo lo que necesitamos proteger en nuestra organización.

1.4.2 Identificación de las amenazas

Una vez que conocemos los recursos que debemos proteger, es la hora de identificar las vulnerabilidades y amenazas que se ciernen contra ellos. Una vulnerabilidad es cualquier situación que pueda desembocar en un problema de seguridad, y una amenaza es la acción específica que aprovecha una vulnerabilidad para crear un problema de seguridad; entre ambas existe una estrecha relación: sin vulnerabilidades no hay amenazas, y sin amenazas no hay vulnerabilidades.

Se suelen dividir las amenazas que existen sobre los sistemas informáticos en tres grandes grupos, en función del ámbito o la forma en que se pueden producir:

- Desastres del entorno.

  Dentro de este grupo se incluyen todos los posibles problemas relacionados con la ubicación del entorno de trabajo informático o de la propia organización, así como con las personas que de una u otra forma están relacionadas con el mismo. Por ejemplo, se han de tener en cuenta desastres naturales (terremotos, inundaciones), desastres producidos por elementos cercanos (como los cortes de fluido eléctrico) y peligros relacionados con operadores (programadores o usuarios del sistema).

- Amenazas en el sistema.

  Bajo esta denominación se contemplan todas las vulnerabilidades de los equipos y su *software* que pueden acarrear amenazas a la seguridad, como fallos en el sistema operativo, medidas de protección que éste ofrece, fallos en los programas, copias de seguridad.

- Amenazas en la red.

Cada día es menos común que una máquina trabaje aislada de todas las demás; se tiende a comunicar equipos mediante redes locales, intranets o la propia Internet, y esta interconexión acarrea nuevas y peligrosas amenazas a la seguridad de los equipos, peligros que hasta el momento de la conexión no se suelen tener en cuenta. Por ejemplo, es necesario analizar aspectos relativos al cifrado de los datos en tránsito por la red, a proteger una red local del resto de Internet, o a instalar sistemas de autenticación de usuarios remotos que necesitan acceder a ciertos recursos internos a la organización (como un investigador que conecta desde su casa a través de un módem).

Algo importante a la hora de analizar las amenazas a las que se enfrentan nuestros sistemas es analizar los potenciales tipos de atacantes que pueden intentar violar nuestra seguridad, esto no es más que el fruto de la repercusión que en todos los medios tienen estos individuos y sus acciones; en realidad, la inmensa mayoría de problemas de seguridad vienen dados por atacantes internos a la organización afectada. En organismos universitarios estos atacantes suelen ser los propios estudiantes (rara vez el personal), así como piratas externos a la entidad que aprovechan la habitualmente mala protección de los sistemas universitarios para acceder a ellos y conseguir así cierto status social dentro de un grupo de piratas. Los conocimientos de estas personas en materias de sistemas operativos, redes o seguridad informática suelen ser muy limitados, y sus actividades no suelen entrañar muchos riesgos a no ser que se utilicen nuestros equipos para atacar a otras organizaciones, en cuyo caso a los posibles problemas legales hay que sumar la mala imagen que nuestras organizaciones adquieren.

No siempre hemos de contemplar a las amenazas como actos intencionados contra nuestro sistema, muchos de los problemas pueden ser ocasionados por accidentes, desde un operador que derrama una taza de café sobre una terminal hasta un usuario que tropieza con el cable de alimentación de un servidor y lo desconecta de la línea eléctrica, pasando por temas como el borrado accidental de datos o los errores de programación; decir "no lo hice a propósito" no ayuda nada en estos casos. Por

supuesto, tampoco tenemos que reducirnos a los accesos no autorizados al sistema: un usuario de nuestras máquinas puede intentar conseguir privilegios que no le corresponden, una persona externa a la organización puede lanzar un ataque de negación de servicio contra la misma sin necesidad de conocer ni siquiera un *login* y una contraseña, etc.

1.4.3 Medidas de protección

Tras identificar todos los recursos que deseamos proteger, así como las posibles vulnerabilidades y amenazas a que nos exponemos y los potenciales atacantes que pueden intentar violar nuestra seguridad, hemos de estudiar cómo proteger nuestros sistemas, sin ofrecer aún implementaciones concretas para protegerlos (esto ya no serían políticas sino mecanismos). Esto implica en primer lugar cuantificar los daños que cada posible vulnerabilidad puede causar, teniendo en cuenta las posibilidades de que una amenaza se pueda convertir en realidad. Este cálculo puede realizarse partiendo de hechos sucedidos con anterioridad en nuestra organización, aunque por desgracia en muchos lugares no se suelen registrar los incidentes ocurridos.

La clasificación de riesgos y medidas de protección suele realizarse con base en el nivel de importancia del daño causado y en la probabilidad aproximada de que ese daño se convierta en realidad; se trata principalmente de no gastar más dinero en una implementación para proteger un recurso de lo que vale dicho recurso o de lo que nos costaría recuperarnos de un daño en él o de su pérdida total. Por ejemplo, podemos seguir un análisis, el riesgo de perder un recurso (a la probabilidad de que se produzca un ataque), y le asignamos un valor de 0 a 10 (valores más altos implican más probabilidad); de la misma forma, definimos también de 0 a 10 la importancia de cada recurso, siendo 10 la importancia más alta.

De esta forma podemos utilizar hojas de trabajo en las que, para cada recurso, se muestre su nombre y el número asignado, así como los tres valores anteriores. Evidentemente, los recursos que presenten un riesgo evaluado mayor serán los que

más medidas de protección deben poseer, ya que esto significa que es probable que sean atacados, y que además el ataque puede causar pérdidas importantes. Es especialmente importante un grupo de riesgos denominados inaceptables, aquellos cuyo peso supera un cierto umbral; se trata de problemas que no nos podemos permitir en nuestros sistemas, por lo que su prevención es crucial para que todo funcione correctamente.

Una vez que conocemos el riesgo evaluado de cada recurso es necesario efectuar lo que se llama el análisis de costos y beneficios. Básicamente consiste en comparar el costo asociado a cada problema (calculado anteriormente) con el costo de prevenir dicho problema. El cálculo de este último no suele ser complejo si conocemos las posibles medidas de prevención que tenemos a nuestra disposición: por ejemplo, para saber lo que nos cuesta prevenir los efectos de un incendio en la sala de operaciones, no tenemos más que consultar los precios de sistemas de extinción de fuego, o para saber lo que nos cuesta proteger nuestra red sólo hemos de ver los precios de productos como *routers* que bloqueen paquetes o cortafuegos completos.

No sólo hemos de tener en cuenta el costo de cierta protección, sino también lo que nos puede suponer su implementación y su mantenimiento; en muchos casos existen soluciones gratuitas para prevenir ciertas amenazas, pero estas soluciones tienen un costo asociado relativo a la dificultad de hacerlas funcionar correctamente de una forma continúa en el tiempo, por ejemplo, dedicando a un empleado a su implementación y mantenimiento.

Cuando ya hemos realizado este análisis no tenemos más que presentar nuestras cuentas a los responsables de la organización (o adecuarlas al presupuesto que un departamento destina a materias de seguridad), siempre teniendo en cuenta que el gasto de proteger un recurso ante una amenaza ha de ser inferior al gasto que se produciría si la amenaza se convirtiera en realidad. Hemos de tener siempre presente que los riesgos se pueden minimizar, pero **nunca** eliminarlos completamente, por lo que será recomendable planificar no sólo la prevención ante de un problema sino también la recuperación si el mismo se produce; se suele hablar de medidas

proactivas (aquellas que se toman para prevenir un problema) y medidas **reactivas** (aquellas que se toman cuando el daño se produce, para minimizar sus efectos).

1.4.4 Estrategias de respuesta

¿Qué hacer cuando nuestra política de seguridad ha sido violada? La respuesta a esta pregunta depende completamente del tipo de violación que se haya producido, de su gravedad, de quién la haya provocado, de su intención. Si se trata de accidentes o de problemas poco importantes suele ser suficiente con una reprimenda verbal o una advertencia, si ha sido un hecho provocado, quizás es conveniente emprender acciones algo más convincentes, como la clausura de las cuentas de forma temporal o pequeñas sanciones administrativas. En el caso de problemas graves que hayan sido intencionados interesará emprender acciones más duras, como cargos legales o sanciones administrativas firmes.

Una gran limitación que nos va a afectar mucho es la situación de la persona o personas causantes de la violación con respecto a la organización que la ha sufrido. En estos casos se suele diferenciar entre usuarios internos o locales, que son aquellos pertenecientes a la propia organización, y externos, los que no están relacionados directamente con la misma; las diferencias entre ellos son los límites de red, los administrativos, los legales o los políticos. Evidentemente es mucho más fácil buscar responsabilidades ante una violación de la seguridad entre los usuarios internos, ya sea contra la propia organización o contra otra, pero utilizando los recursos de la nuestra.

Existen dos estrategias de respuesta ante un incidente de seguridad.

- Proteger y proceder.
- Perseguir y procesar.

La primera de estas estrategias, proteger y proceder, se suele aplicar cuando la organización es muy vulnerable o el nivel de los atacantes es elevado; la filosofía es proteger de manera inmediata la red y los sistemas y restaurar su estado normal, de forma que los usuarios puedan seguir trabajando normalmente. Seguramente será necesario interferir de forma activa las acciones del intruso para evitar más accesos, y analizar el daño causado. La principal desventaja de esta estrategia es que el atacante se da cuenta rápidamente de que ha sido descubierto, y puede emprender acciones para ser identificado, lo que incluso conduce al borrado de *logs* o de sistemas de ficheros completos; incluso puede cambiar su estrategia de ataque a un nuevo método, y seguir comprometiendo al sistema. Sin embargo, esta estrategia también presenta una parte positiva: el bajo nivel de conocimientos de los atacantes en sistemas habituales hace que en muchas ocasiones se limiten a abandonar su ataque y dedicarse a probar suerte con otros sistemas menos protegidos en otras organizaciones.

La segunda estrategia de respuesta, perseguir y procesar, adopta la filosofía de permitir al atacante proseguir sus actividades, pero de forma controlada y observada por los administradores, de la forma más discreta posible. Con esto, se intentan guardar pruebas para ser utilizadas en la segunda parte de la estrategia, la de acusación y procesamiento del atacante (ya sea ante la justicia o ante los responsables de la organización, si se trata de usuarios internos). Evidentemente corremos el peligro de que el intruso descubra su monitorización y destruya completamente el sistema, así como que nuestros resultados no se tengan en cuenta ante un tribunal debido a las artimañas legales que algunos abogados aprovechan; la parte positiva de esta estrategia es, aparte de la recolección de pruebas, que permite a los responsables conocer las actividades del atacante, qué vulnerabilidades de nuestra organización ha aprovechado para atacarla, cómo se comporta una vez dentro, etc. De esta forma podemos aprovechar el ataque para reforzar los puntos débiles de nuestros sistemas.

A nadie se le escapan los enormes peligros que entraña el permitir a un atacante proseguir con sus actividades dentro de las máquinas; por muy controladas que estén,

en cualquier momento casi nada puede evitar que la persona se sienta vigilada, se ponga nerviosa y destruya completamente nuestros datos. Una forma de monitorizar sus actividades sin comprometer excesivamente nuestra integridad es mediante un proceso denominado *jailing* o encarcelamiento: la idea es construir un sistema que simule al real, pero donde no se encuentren datos importantes, y que permita observar al atacante sin poner en peligro los sistemas reales. Para ello se utiliza una máquina, denominada **sistema de sacrificio**, que es donde el atacante realmente trabaja, y un segundo sistema, denominado **de observación**, conectado al anterior y que permite analizar todo lo que esa persona está llevando a cabo. De esta forma conseguimos que el atacante piense que su intrusión ha tenido éxito y continué con ella mientras lo monitorizamos y recopilamos pruebas para presentar en una posible demanda o acusación. Si deseamos construir una cárcel es necesario que dispongamos de unos conocimientos medios o elevados de programación de sistemas.
Sin importar la estrategia adoptada ante un ataque, siempre es recomendable ponerse en contacto con entidades externas a nuestra organización, como el CERT.

El análisis de riesgos no debe ser hecho una sola vez y olvidado, debe ser actualizado periódicamente. Además el análisis de amenazas debe ser realizado cada vez que se observe un cambio importante en la operación o la estructura del inmueble, por ej., un cambio de oficinas, la construcción de una oficina cerca de nuestro centro de cómputo, remodelaciones, etc.

Todo esto nos servirá para realizar las adecuaciones y correcciones a cada situación que lo necesite, para ello se utilizarán mecanismos o herramientas propias para cada caso, por ejemplo para el riesgo de la seguridad en control de acceso en los sistemas podrían ser:

- Herramientas de monitoreo
- Ruteadores con filtrado
- Firewall

Los cuales serán una alternativa para dichas adecuaciones y/o correcciones a nuestras necesidades.

CAPÍTULO

2

GENERALIDADES Y ARQUITECTURA DE FIREWALLS

2. Generalidades y Arquitectura de Firewalls

La seguridad puede ser soportada por diversos esquemas y herramientas, una de ellas son los Firewall, se hablará sobre lo que es y qué puede hacer o no con él, así como las distintas arquitecturas sobre estos, además de filtrado de paquetes.

2.1 Definiciones de Firewalls

No existe un definición precisa de *Firewalls*, por lo que aquí se presenta algunas que se relacionan a la descripción y funcionamiento de un *Firewall*.

Firewall

Definición 1:

"Un componente o conjunto de componentes que restringen el acceso entre una red protegida e Internet, o entre otros conjuntos de redes"[9]

Definición 2:

Los Firewall *(firewalls* o *networtk firewalls* en la literatura anglosajona) son dispositivos o sistemas que controlan el flujo de tráfico entre dos o más redes empleando ciertas políticas de seguridad. Básicamente son dispositivos cuya funcionalidad se limita a permitir o bloquear el tráfico entre dos redes en base a una serie de reglas. Su complejidad reside en las reglas que admiten y en cómo realizan la toma de decisiones en base a dichas reglas.[10]

[9] BRENT, Chapman, ZWICKY Elizabeth. *"Construya Firewalls para Internet"*, O'Reilly, México, *1997.*

[10] "Firewalls", http://jo.morales0002.eresmas.net/fencasa.html Pág. 7, consultada en 2003.

Definición 3:

Firewall o cortafuegos es un sistema o grupo de sistemas que hace cumplir una política de control de acceso entre dos redes. De una forma más clara, se puede definir un Firewall como cualquier sistema (desde un simple *router* hasta varias redes en serie) utilizado para separar en cuanto a seguridad se refiere una máquina o *subred* del resto, protegiéndola así de servicios y protocolos que desde el exterior puedan suponer una amenaza a la seguridad. El espacio protegido, denominado perímetro de seguridad, suele ser propiedad de la misma organización, y la protección se realiza contra una red externa, no confiable, llamada zona de riesgo.

Definición 4:

En términos de seguridad informática se puede hacer una analogía entre el cortafuegos (Firewall) de una red o de una computadora y los muros que rodean a un castillo medieval. Los muros protegen las construcciones interiores del castillo, con acceso a la ciudad a través de una puerta controlada, reforzada para aguantar el asalto.

Como ha notado, las definiciones abarcan los mismos puntos, definen a un Firewall como un sólo elemento o hasta un conjunto de elementos que permitirán tener aislada nuestra red local de la red externa, así como de los peligros que abarca ésta, particularmente la definición 3 es la que abarca los puntos esenciales de lo que es un Firewall.

Existen muchos tipos de Firewall, no obstante la clasificación más clara quizá sería la que los diferencia según la forma de implementar la política de seguridad del sitio atendiendo al nivel de la capa OSI en la que se implementa dicha política de seguridad.

- En un primer lugar existen los Firewall de nivel 3 de la capa OSI, esto es, de nivel de red o, lo que es lo mismo, nivel IP en redes TCP/IP como Internet. Estos Firewall pueden ser considerados como filtros de paquetes, ya que lo que realizan, a fin de cuentas, es un filtrado de los intentos de conexión atendiendo a direcciones IP origen y destino y puerto de destino de los paquetes IP. También

se podrá especificar desde qué direcciones IP origen dará acceso a los servidores públicos.

- Otra posibilidad de implementación de Firewall es a nivel 4 de OSI, esto es a nivel de transporte o de TCP en redes TCP/IP. En este nivel ya se puede atender a aspectos de si los paquetes son de inicio de conexión o se corresponden con paquetes cuyas conexiones están ya establecidas.
- Por último nos quedan los Firewall a nivel 7 de la capa OSI, esto es, a nivel de aplicación. Estos Firewall actúan a modo de *proxy* para las distintas aplicaciones que van a controlar.

"Las terminologías sobre Firewalls difieren. Mientras algunos autores utilizan el término servicio *proxy* para abarcar todo el enfoque *proxy*, otros autores se refieren a compuertas a nivel de aplicación y compuertas a nivel de circuito.[11]"

Con base en esta diferencia se presenta un cuadro de ventajas y desventajas de ambas clasificaciones:

TECNOLOGÍA	VENTAJAS	INCONVENIENTES
Filtro de paquetes	Alto Rendimiento Simplicidad	Conexión directa Ataques sofisticados
Pasarela de Aplicación (*Proxy*)	Aislamiento de la conexión entre redes Alta Seguridad	Retraso en implantación nuevas aplicaciones Menor Rendimiento

Tabla 3. Tecnologías de Firewalls

[11] BRENT, Chapman, ZWICKY Elizabeth. *"Construya Firewalls para Internet"*, O'Reilly, México, 1997, p 61.

La figura siguiente muestra la forma más simple de representar un Firewall.

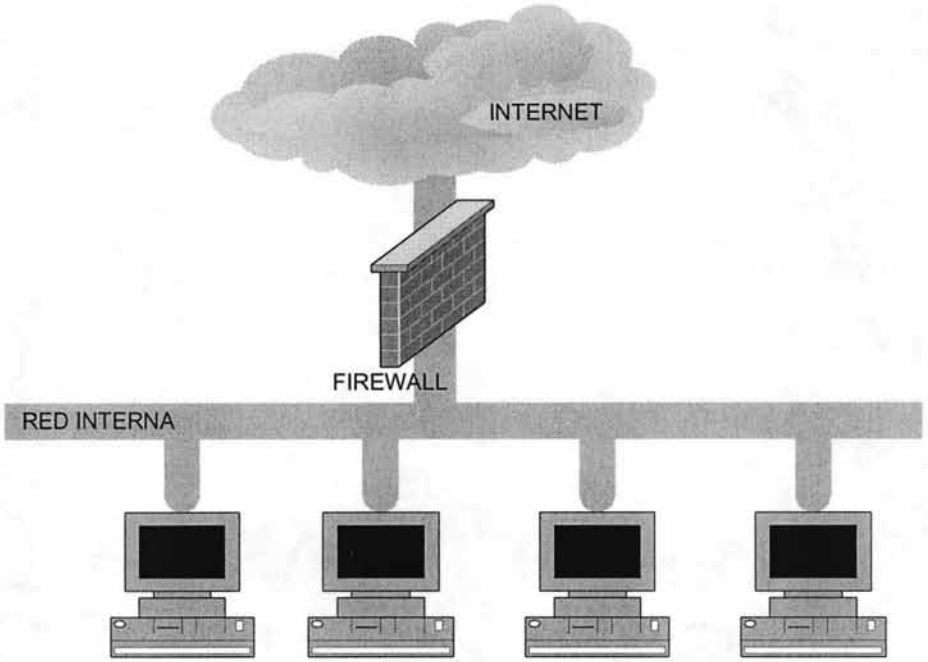


Fig. 2. Un Firewall por lo general separa una red interna de Internet

2.2 ¿Qué puede y qué no puede hacer un Firewall?

Tradicionalmente se ha dicho en los ámbitos de seguridad en redes que la máquina realmente segura es aquella que esta físicamente desconectada de todas las redes internas o externas (sin ninguna tarjeta de red), tampoco tendrá unidad de disco (y en su caso ninguna unidad de almacenamiento externo, como grabadores y lectores de cintas, discos extraíbles, etc.) o acceso a impresoras y se encontrará en una habitación acorazada con un guardia insobornable, y si está máquina está apagada que mejor. Pero para casos prácticos esto no es posible por lo que se debe buscar mecanismos que ayuden a disminuir los riesgos, pero sabiendo que son parte de todo un sistema de seguridad.

Un *Firewall* nunca protegerá al cien por ciento a estos recursos internos de acceso no autorizados ya que las técnicas de intrusión avanzan día a día (aunque el hardware y software de los *Firewall* también) y todos los días se descubren nuevos fallos (*bugs*) en sistemas operativos y software de servidores. Pero también es cierto que un *Firewall* bien configurado junto con servidores bien configurados y protegidos puede poner las cosas muy difíciles a estos potenciales intrusos.

El nivel de protección que puede darnos un *Firewall* depende en gran medida, de nuestras necesidades. Imaginemos, por ejemplo, que nuestra única necesidad es recibir y entregar correos electrónicos. Un *Firewall* puede defendernos efectivamente de cualquier ataque que no vaya dirigido a este servicio.

Generalmente, los *Firewalls* se configuran para proteger contra cualquier intento de acceso no autorizado o no correctamente autenticado desde el exterior hacia el interior de nuestra red, o viceversa. Pero, adicionalmente uno de los puntos más importantes a tener en cuenta es que un *Firewall* proporciona un punto único e ineludible de acceso a la red donde podrá centralizar las medidas de seguridad y auditoria sobre la misma.

Son tres las principales amenazas sobre las cuales un Firewall no puede proteger:

- Un Firewall no puede proteger contra amenazas que no pasan a través de él. Como decía en el punto anterior, el Firewall debe de ser el punto único e ineludible de acceso a nuestra red. Si esto no es así su efectividad es sólo parcial.

- Tampoco pueden proteger, generalmente, contra amenazas que proceden del interior de nuestra red. Un empleado malicioso, un troyano o algunos tipos de virus pueden usar mecanismos válidos 'desde dentro' para realizar acciones perniciosas.

- Finalmente, los Firewalls no pueden proteger contra clientes o servicios que de admitan como válidos, pero que son vulnerables. Tampoco puede proteger contra mecanismos de *tunneling* sobre HTTP, SMTP u otros protocolos. No son muy efectivos, a pesar de que algunos fabricantes así lo anuncian, contra los virus. Los Firewalls no pueden ni deben sustituir otros mecanismos de seguridad que reconozcan la naturaleza y efectos de los datos y aplicaciones que se estén manejando y actúen en consecuencia.

No se debe olvidar tomar otras medidas de seguridad para cada máquina, alguna será instalar Tcp Wrapers entre otras herramientas de seguridad como las descritas en el apéndice B, las máquinas Linux ya tienen TCP Wrapers, pero en el Departamento se maneja también máquinas SUN con Sistema Operativo Solaris el cual no lo tiene instalado, por lo tanto, se instalará en las máquinas con este Sistema Operativo (en el apéndice B se describe esta herramienta junto con otras).

Qué puede hacer un Firewall	**Qué no puede hacer**
Aislar una red interna de una externa.	Proteger de amenazas que no pasan a través de él.
Protege los recursos internos de intentos de acceso no permitidos.	Proteger de fallos (*bugs*) en el sistema operativo o en software.
Filtra paquetes no deseados.	Fallas de hardware.
Da un nivel de protección con base en nuestras necesidades.	Si no es el único punto de acceso a la red, su efectividad es sólo parcial.
Siendo el punto único e ineludible de acceso a la red se pueden centralizar medidas de seguridad y auditoria sobre el sistema.	No protege de amenazas procedentes del interior de la red (troyanos, virus, empleados maliciosos, etc.).
	No protege contra clientes o servicios que admitamos como válidos, pero que son vulnerables.
Un *Firewall* bien configurado protege de ataques que específicamente se nieguen.	No sustituye otros mecanismos de seguridad.
	No protege de daños físicos en contra de terremotos, inundaciones, etc.
Dar un cierto grado de seguridad.	No protege al 100 %.

Tabla 4. Resumen de lo qué puede y no hacer un Firewall.

Mucho de lo que puede y no hacer un Firewall por usted dependerá de la forma que se implementa, con base en otros diseños o arquitectura ya establecida, o bien, después de hacer un análisis para diseñar un Firewall propio.

2.3 Arquitectura de Firewalls

En este apartado se hablará de las posibles configuraciones para montar un Firewall y en otros apartados se definirán algunos de los componentes que lo integran. Un concepto básico en este apartado es: Anfitrión *bastión*, por lo que a continuación se explicará el significado de éste.

Anfitrión Bastión

El anfitrión *bastión* se refiere a su presencia pública en Internet y se encuentra expuesto a elementos potencialmente hostiles, es en este sistema al que los extraños deberán conectarse, por lo general, para tener acceso a un servicio de comercio electrónico (o de otra índole). Por lo tanto, será en estas máquinas donde se instalará y hará funcionar los servidores de *web*.

Por su diseño el *bastión* es una máquina muy expuesta, por lo que es necesario concentrar los esfuerzos de seguridad en torno a él, en nuestro caso doblemente, pues un acceso no autorizado a los recursos del *bastión* nos puede dar un sinfín de problemas, desde perder datos (y recibir una multa) hasta ser víctimas de un ataque al prestigio de nuestro sitio.

host bastión, el término *host bastión* se atribuye a Marcus Ranum Ranum:

> "Los *bastiones*... tienen una vista dominante de las áreas criticas de defensa, normalmente de muros más poderosos, espacio para más tropas y, de cuando en cuando, se vierte aceite hirviendo para disuadir a los atacantes"[12].

2.3.1 Arquitectura de anfitrión con doble acceso

La forma más simple de Firewall es el sistema de *hosts bastión* con dos interfaces de red o anfitrión *bastión* como lo manejaremos aquí.

Se construye alrededor de una computadora con al menos dos interfaces de red. Tal máquina podría actuar como *router* entre las redes a las que están conectadas sus tarjetas; es capaz de enrutar paquetes IP de una red a otra. Sin embargo, al implementar una arquitectura de Firewalls de tipo máquina con doble acceso, desactive esta función como *router*. Así, los paquetes IP de una red no serán enrutados de forma directa a la otra.

Los sistemas dentro del Firewall pueden comunicarse con la máquina con doble acceso y los sistemas fuera del Firewall pueden comunicarse con la máquina con doble acceso, pero los sistemas no pueden comunicarse entre sí de manera directa. El tráfico IP entre ellos está bloqueado.

La arquitectura de red para un Firewall de máquina con doble acceso es bastante sencilla, el anfitrión se coloca entre Internet y la red interna.

---

[12] BRENT, Chapman, ZWICKY Elizabeth. *"Construya Firewalls para Internet"*, O'Reilly, México, *1997*, p. 58.

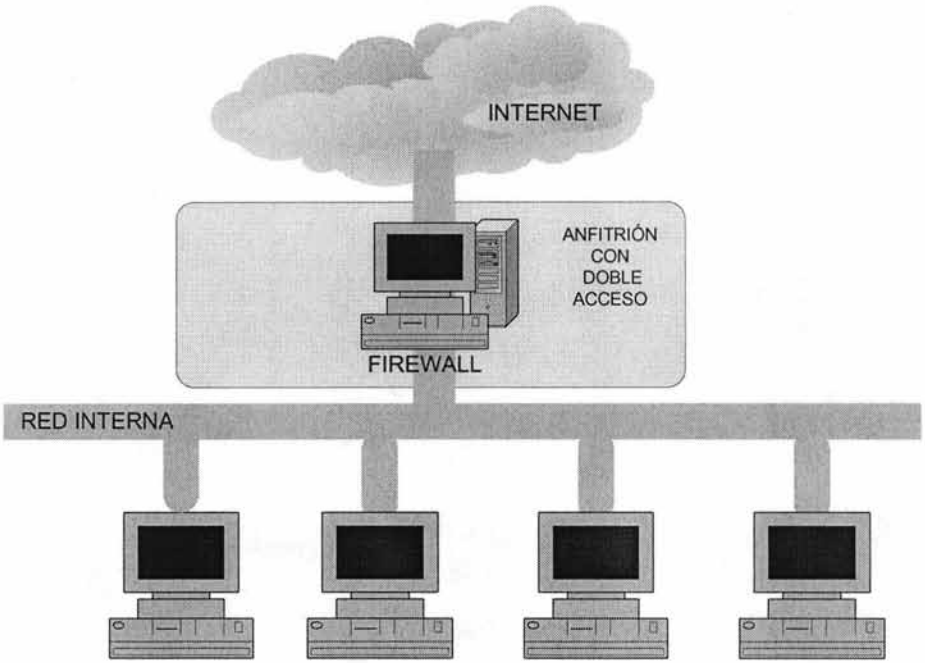


Fig. 3. ARQUITECTURA DE ANFITRIÓN CON DOBLE ACCESO

Este sistema puede proporcionar un alto nivel de control, en algunos casos permitiremos que rechace conexiones que pretenden ser para un servicio específico, pero que en realidad, no contienen el tipo correcto de datos.

Una máquina con doble acceso sólo puede proporcionar servicios tipo *proxy* o hacer que los usuarios inicien una sesión directa con él. Las cuentas de usuarios presentan problemas de seguridad importantes por sí mismas. Presentan problemas especiales en máquinas con doble acceso, donde pueden inesperadamente activar servicios que podrían ser inseguros. Además que la mayoría de los usuarios encuentran que es inconveniente emplear una máquina con doble acceso iniciando con ella una sesión antes de salir a Internet.

2.3.2 Arquitectura de anfitrión de protección

Esta arquitectura proporciona servicios en un anfitrión conectado sólo a la red interna, utilizando un *router* independiente, la seguridad principal de esta arquitectura la proporciona el filtrado de paquetes.

El anfitrión *bastión* está colocado en la red interna. El filtrado de paquetes en el *router* de protección está configurado de tal manera que el *bastión* es el único sistema en la red interna con el que los ordenadores exteriores pueden abrir conexiones. Sólo están permitidas ciertos tipos de conexiones. Cualquier sistema que intente tener acceso a los sistemas o servicios internos tendrá que conectarse con este *bastión*, por lo tanto, éste deberá tener el nivel más alto de seguridad disponible. El filtrado permite que el *bastión* abra conexiones específicas al mundo exterior.

La configuración para el filtrado de paquetes en el *router* puede:

- Permitir que se abran conexiones con Internet para ciertos servicios, desde dentro hacia fuera.
- No permitir todas las conexiones de anfitriones internos (obligando a utilizar servicios *proxy* para realizar sólo conexiones seguras).

La Figura 4 muestra una versión sencilla de una arquitectura de anfitrión de protección.

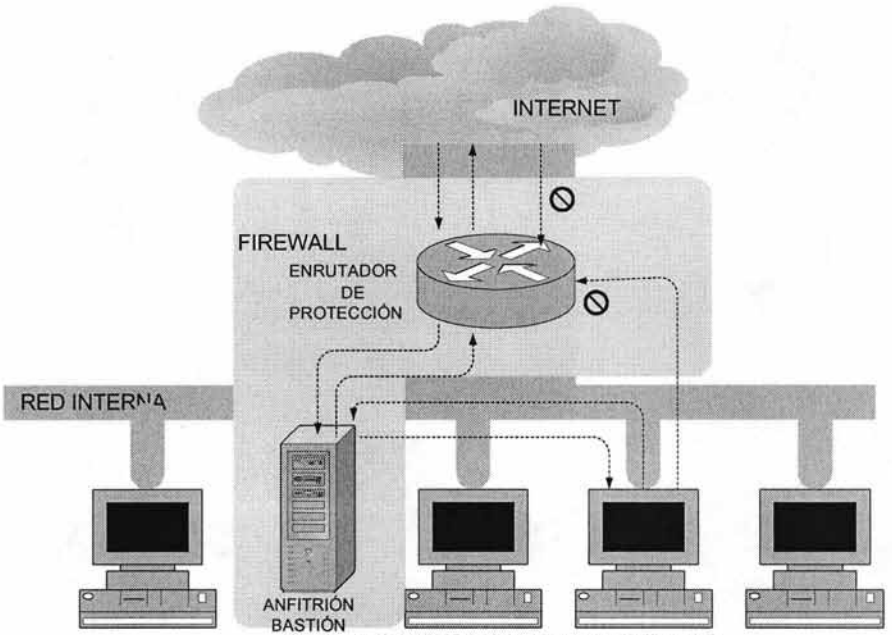


Fig. 4. ARQUITECTURA DE ANFITRIÓN DE PROTECCIÓN

Debido a que esta arquitectura permite que los paquetes se muevan de Internet a las redes internas, pueden parecer más riesgoso que la arquitectura de anfitrión con doble acceso, diseñada para que ningún paquete externo alcance la red interna, sin embargo esta arquitectura también está propensa a fallas.

Es más fácil defender el *bastión*, el cual proporciona un conjunto muy limitado de servicios, que defender un anfitrión interno. Para casi todos los propósitos la arquitectura de *bastión* de protección proporciona mejor seguridad y uso que la de doble acceso.

Sin embargo, tiene desventajas: si un ataque puede penetrar en el *bastión* no queda nada por detrás que sirva como protección.

2.3.3 Arquitectura de subred de protección

Esta arquitectura agrega una capa adicional de seguridad a la arquitectura de anfitrión de protección al añadir una red de perímetro que aísla aún más la red interna de Internet, como se muestra en la Figura 5.

Como en algunas batallas la línea de defensa no está constituida por una sola trinchera (el *bastión*), sino por una red de trincheras que se adentran en terreno amigo; en este caso esta red de trincheras la constituirá una subred (informática) de protección.

En la forma más sencilla de subred de protección, hay dos *routers*, cada uno conectado a un lado del *bastión*. Algunas instalaciones van más lejos en la creación de subredes de protección, configurando una serie de capas de redes de perímetro alrededor de la red interna.

Los servicios menos fiables y vulnerables se colocan en las redes de perímetro exteriores, más lejos de la red interior. La idea es que al atacante que penetre con éxito una máquina de la red exterior, le cueste cada vez más atacar las máquinas internas debido a las capas de seguridad entre el perímetro exterior y la red interna. Sin embargo esto sólo sucederá si las distintas capas tienen sentido; si los demás sistemas de filtrado entre cada capa permiten lo mismo entre todas las capas, éstas adicionales no proporcionarán mayor seguridad.

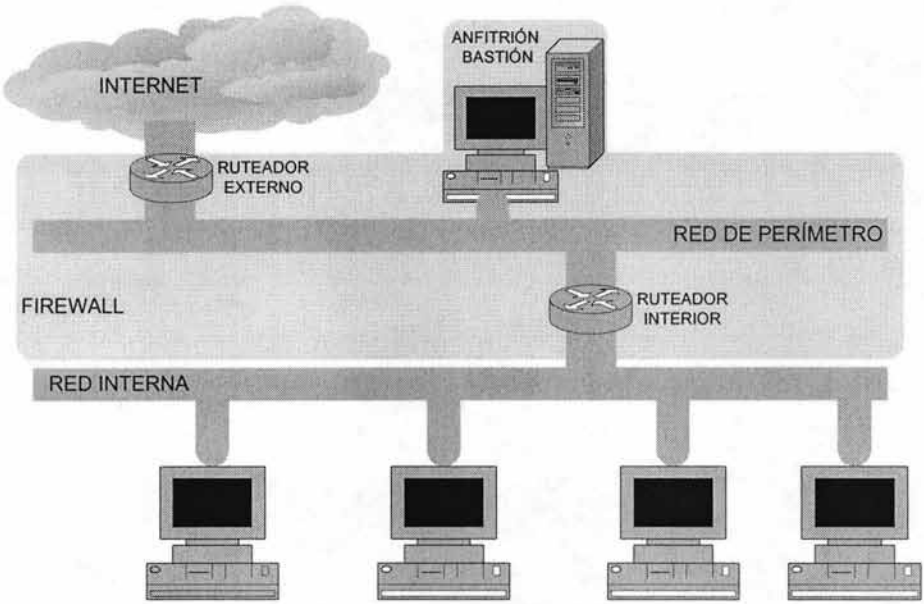


Fig. 5. ARQUITECTURA DE SUBRED DE PROTECCIÓN(utilizando dos ruteadores)

2.3.4 Componentes de los Firewalls y variaciones en las arquitecturas

He mostrado las arquitecturas de Firewalls más comunes, en la figuras 3, 4 y 5, y se observan diversas variantes entre ellas. Existe una gran flexibilidad en la configuración y combinación de los componentes de un Firewall para adaptarse mejor al presupuesto y política de seguridad, por lo que a continuación se mostrará algunas variantes de las arquitecturas y descripción de algunos componentes.

Red de perímetro

Es una capa de seguridad extra, una red adicional entre la red Internet y la red protegida. Si un intruso entra a los límites externos del Firewall, la red de perímetro ofrece una capa adicional de protección entre el atacante y los sistemas internos.

En muchas configuraciones de red, es posible para cualquier máquina en una red determinada escuchar todo el tráfico de todas las máquinas (Ethernet, Token Ring y FDDI), los atacantes pueden obtener contraseñas y otros datos vulnerables analizando este tráfico.
Debido a que este tráfico estrictamente interno no pasará por la red de perímetro, estará seguro de atacantes externos.

Router interior

Protege la red interna tanto de Internet como de la red de perímetro. Realiza la mayor parte del filtrado de paquetes para el Firewall. Permite que los servicios seleccionados salgan de la red interna hacia Internet.

Los servicios que permite el *router* interno entre su *bastión* y la red interna no son necesariamente los mismos que se permite entre Internet y la red interna.

El objetivo de limitar los servicios entre el *bastión* y la red interna es reducir el número de máquinas que pueden ser atacadas desde el *bastión* si éste está comprometido en un ataque. Deberá limitar los servicios permitidos entre el *bastión* y la red interna sólo a los que en realidad necesite, como SMTP y/o DNS. Se debe limitar mucho más los servicios, haciendo que únicamente sean posibles hacia y desde determinadas máquinas de la red interna.

Router exterior

Protege tanto la red interna como la de perímetro contra Internet, tiende a permitir casi cualquier cosa que salga de la red perímetro y, en general, realizan poco filtrado.

Las únicas reglas de filtrado especiales, son las que protegen al *bastión* y el *router* interno. El resto de reglas, evitan que el tráfico inseguro pase entre los anfitriones

internos e Internet. Para soportar el servicio de *proxy*, donde el *router* interno permita que las máquinas internas envíen algunos protocolos siempre y cuando hablen con el *bastión*, el *router* externo podría dejar pasar estos protocolos siempre y cuando vengan sólo del *bastión*. Estas reglas adicionales son en teoría innecesarias y redundantes, pero proporcionan un nivel de seguridad adicional muy deseable.

Así pues ¿que necesita hacer el *router* exterior?, una de las tareas es el bloqueo de cualquier paquete que entre de Internet con direcciones fuentes falsificadas, éstas dicen venir de la red interna pero lo hacen por la interfaz de Internet.

Replicar bastiones.

Aunque se hable de un sólo *bastión*, es perfectamente viable y además incluso deseable replicar éste, de manera que separemos servicios o aumentemos el nivel de redundancia. Aunque deberemos tener cuidado puesto que no todos los servicios soportan este enfoque.

Fusionar Routers interno y externo.

Podemos fusionar los *routers* interno y externo únicamente si se tiene un *router* con suficiente capacidad y flexibilidad, tal que permite especificar tanto los filtros de entrada como los de salida en cada interfaz.

Fusionar el anfitrión bastión y el ruteador exterior.

Quizá haya casos en que se utilice una sola máquina con doble acceso como anfitrión *bastión* y como *ruteador* exterior. El uso de un anfitrión con doble acceso para enrutar tráfico no le dará el rendimiento o la flexibilidad de un *ruteador* dedicado, pero no necesita mucho de ambos para una sola conexión de bajo ancho de banda.
A diferencia de fusionar los *ruteadores* interior y exterior, al fusionar el anfitrión *bastión* con el *ruteador* exterior, no se crea nuevas vulnerabilidades significativas. Lo que sí

hace es exponer aún más al anfitrión *bastión*. En esta arquitectura, el anfitrión *bastión* está más expuesto a Internet, protegido sólo por cualquier filtrado (si lo hay), por lo que necesita cuidados adicionales para protegerlo.

Fusionar el anfitrión bastión y el ruteador interior.

Aunque es aceptable fusionar el anfitrión *bastión* y el *ruteador* exterior, como se mencionó antes; no es buena idea fusionar el anfitrión *bastión* y el *ruteador* interno, el hacerlo pone en riesgo la seguridad en general.

El anfitrión *bastión* y el *ruteador* exterior realizan cada uno tareas de protección distintas; se complementan pero no se respaldan entre sí. El *ruteador* interior funciona, en parte como respaldo de los dos.

Uno de los propósitos principales de la red de perímetro es evitar que el anfitrión *bastión* curiosee el tráfico interno. Mover el anfitrión *bastión* al *ruteador* interior hace que todo el tráfico sea visible para (o desde) él.

Existen más variantes las cuales tienen ventajas y desventajas como las anteriores, la elección de alguna de estas se basará en el presupuesto y la política que se maneje.

Se muestra un resumen de las arquitecturas en la siguiente tabla:

ARQUITECTURAS

Anfitrión con doble acceso	Anfitrión de protección	Subred de protección
- Forma más simple de Firewall, compuesto por una computadora y dos interfaces de red. - Se conecta entre Internet y la red interna, pudiendo actuar como *router* entre las redes que están conectadas a sus tarjetas. - Los sistemas dentro del Firewall pueden comunicarse con la máquina con doble acceso, y los sistemas fuera del Firewall pueden comunicarse con la máquina con doble acceso, pero los sistemas no pueden comunicarse entre si de manera directa. El tráfico IP entre ellos está bloqueado. - Proporciona un alto nivel de control. - Proporciona solo servicios tipo *proxy*, o hacer que los usuarios inicien una sesión directa con él. - La mayoría de los usuarios encuentran que es inconveniente emplear una máquina con doble acceso iniciando con ella una sesión antes de salir a Internet.	- Esta arquitectura proporciona servicios en un anfitrión conectado sólo a la red interna, utilizando un *router* independiente, la seguridad principal de ésta arquitectura la proporciona el filtrado de paquetes - El anfitrión *bastión* ésta colocado en la red interna. - El *bastión* es el único sistema en la rede interna con el que los ordenadores exteriores pueden abrir conexiones. - Permitidos solo ciertos tipos de conexiones. - Permite que los paquetes se muevan de Internet a las redes internas por medio de filtrado de paquetes. - Es más fácil defender el *bastión* ya que proporciona un conjunto muy limitado de servicios, que defender un anfitrión interno. - Esta arquitectura proporciona mejor seguridad y uso que la de doble acceso. - Si un ataque puede penetrar en el *bastión*, no queda nada por detrás que sirva como protección.	- Agrega una capa adicional de seguridad a la arquitectura de anfitrión de protección al añadir una red de perímetro. - Aísla más a la red interna. - Constituida por una red de trincheras (subred informática de protección). - En la forma más sencilla de subred de protección, hay dos *routers*, cada uno conectado a un lado del *bastión*. Algunas instalaciones van más lejos en la creación de subredes de protección, configurando una serie de capas de redes de perímetro alrededor de la red interna. - Los servicios menos fiables y vulnerables se colocan en las redes de perímetro exteriores, lejos de la red interior. - La idea es que al atacante que penetre con éxito una máquina de la red exterior, le cueste cada vez más atacar las máquinas internas debido a las capas de seguridad entre el perímetro exterior y la red interna.

Tabla 5. Resumen de arquitecturas de Firewalls

2.4 Filtrado de paquetes

El término Firewall tiene varios significados que dependen del mecanismo que se use para implementarlo, el nivel de la pila de TCP/IP sobre el que funciona el Firewall y las arquitecturas de red y enrutamiento que se usen, uno de los significados más comunes se refieren a un Firewall de filtrado de paquetes.

Un Firewall de filtrado de paquetes se suele implementar dentro del sistema operativo y funciona en las capas de transporte y red TCP/IP como se muestra en la Figura 6. Protege el sistema realizando las decisiones del enrutamiento después de filtrar los paquetes basándose en la información del encabezado del paquete IP.

Un Firewall de filtrado de paquetes consta de una lista de reglas de aceptación y de negación. Estas reglas definen explícitamente los paquetes que se permiten pasar y los que no a través de la interfaz de red. Las reglas del Firewall usan los campos del encabezado del paquete de TCP IP para decidir si enrutar un paquete a su destino, eliminar el paquete o bloquear un paquete y devolver una condición de error a la máquina emisora.

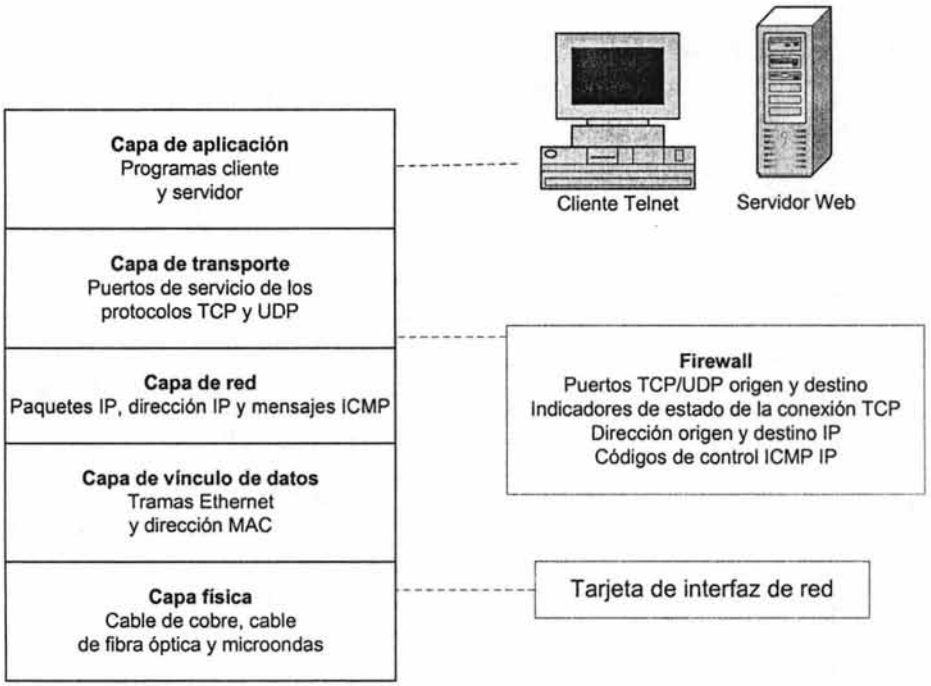


Fig 6 Ubicación del Firewall en el modelo de referencia TCP/IP

Estas reglas se basan en la tarjeta de interfaz de red específica y en la dirección IP del *host*. Las direcciones IP origen y destino del nivel de red, los puertos de servicio UDP y TCP de la capa de transporte, los indicadores de conexión TCP, los tipos de mensaje ICMP del nivel de red y si el paquete es entrante o saliente.

Las listas de reglas que definen lo que puede entrar y lo que puede salir se llaman cadenas porque compara un paquete de cada regla de la lista, una a una hasta que encuentra una coincidencia o la lista termina, como se muestra en la Figura 7.
Cada cadena del Firewall tiene una directiva predeterminada y una colección de acciones a realizar en respuesta a tipos de mensajes específicos. Cada paquete se

compara uno a uno, con cada regla de la lista hasta que se encuentra una coincidencia. Si el paquete no coincide con ninguna regla, fracasa y se aplica la directiva predeterminada al paquete.

Hay dos perspectivas básicas para un Firewall:

- "No todo lo específicamente permitido está prohibido" (Negación preestablecida)
- "No todo lo específicamente prohibido está permitido" (Permiso preestablecido)

La negación preestablecida tiene sentido desde el punto de vista de la seguridad, acepta que lo que no conocemos puede dañarnos, es la opción más segura para muchas personas aunque para los usuarios no mucho.

Con esta postura prohibe por omisión y después para permitir algo debe:

- Examinar los servicios que necesiten los usuarios.
- Considerar cómo afectarían la seguridad tales servicios y cómo puede proporcionarlos de forma segura.
- Permitir sólo los servicios que comprende y que puede proporcionar en forma segura y que sean necesarios.

El permiso preestablecido es el preferido de los usuarios ya que les permite utilizar más servicios, pero esto supone que conoce de manera precisa todos los servicios y cuáles son los peligros específicos, obviamente esto no es muy cierto, por lo que en el Departamento de Administración de Servidores optaremos por la primera postura, aunque esto no quiere decir que no se pueda utilizar la segunda, lo cual se dejará a decisión de cada administrador.

2.4.1 ¿Por qué filtrado de paquetes?

Este sistema nos permite de una forma automática controlar la transferencia de información con base en:

- Dirección origen del paquete.
- Dirección destino.
- Protocolos de nivel sesión y aplicación utilizados

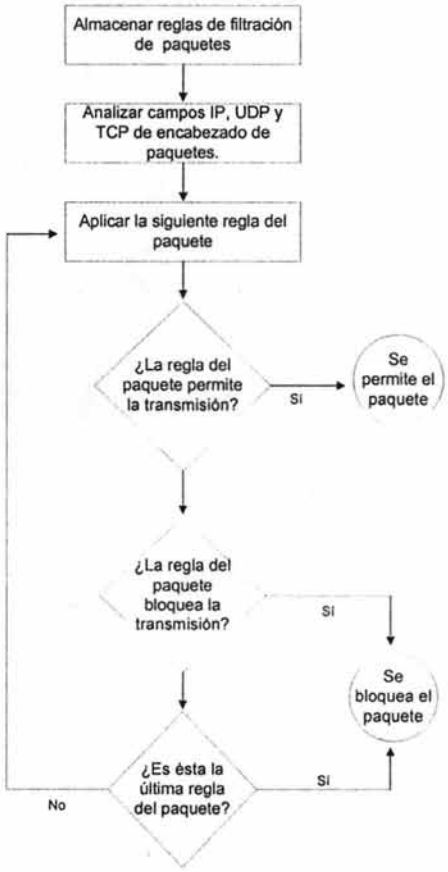


Fig 7 Diagrama de flujo de la operación de filtrado de paquetes

La mayoría de los sistemas para filtrado de paquetes no tomarán decisiones basándose en la propia información ni tampoco en el contenido. El filtrado de paquetes permite decir:

No permita que alguien use FTP acceder al sistema desde el exterior.
Permita a todos conectarse a otros sistemas exteriores con SSH

Pero no le permitirá decir:

El usuario X puede utilizar FTP desde el exterior, pero ningún otro puede hacerlo.

Esto es así porque "usuario X" no es algo que el sistema de filtrado de paquetes pueda identificar.

La principal ventaja del filtrado de paquetes consiste en concentrar, nos permite proporcionar un sólo lugar desde donde ejecutar la protección para toda una red.

2.4.2 Ventajas del filtrado de paquetes.

Un sólo *router* con filtrado colocado estratégicamente puede ayudar a proteger toda una red. No necesita conocimiento o cooperación del usuario, no requiere de ningún programa o configuración especial de las máquinas de los clientes, ni necesita de formación por parte de los usuarios.

El filtrado puede hacerse sin la cooperación y sin el conocimiento de los usuarios, podremos utilizarlo sin tener que molestar a nadie haciéndoles aprender algo nuevo y sin tener que depender de ellos para que funcione.
Esta capacidad está disponible en muchos productos de hardware y software disponibles gratuitamente en Internet.

2.4.3 Desventajas del filtrado de paquetes.

Las herramientas no son perfectas, en mayor o menor grado. La capacidad de filtrado de paquetes en muchos de los productos tienen limitaciones comunes:

- Las reglas de filtrado tienden a ser difíciles de configurar o crear; desde lo difícil con un ligero dolor de cabeza, hasta lo imposible, con una fuerte jaqueca.
- Una vez configuradas, las reglas tienden a ser difíciles de comprobar.
- Las capacidades de filtrado en algunos productos están incompletas.
- Como cualquier otro producto pueden tener problemas, pero estos tienden a convertirse en graves problemas de seguridad.

Los protocolos no están diseñados para el filtrado de paquetes, aun con usos perfectos del filtrado, determinados protocolos no "casan" bien con las técnicas de filtrado de paquetes, como por ejemplo los comando r de Berkeley.

Algunas políticas no pueden aplicarse de inmediato por medio de *routers* comunes con filtrado, la información de la cual dispone un *router* no permite especificar ciertas reglas, como por ejemplo el usuario que está detrás de la transmisión.

Configuración de routers con sistema de filtrado.

Previo a cualquier otro paso deberá decidir que servicios quiere permitir o negar, y después convertir estas decisiones en reglas correctas, generalmente, los protocolos son bidireccionales; casi siempre incluyen un extremo que envía la solicitud y otro que contesta. Al diseñar las reglas deberá recordar que los paquetes viajan en ambas direcciones (e.g. Telnet).

Tampoco servirá de nada bloquear medias conexiones, es decir, sólo la salida desde nuestra red, ya que muchos ataques (por no decir todos) se pueden realizar si el atacante puede introducir paquetes en nuestra red, aún sin obtener respuestas

correctas. Deberá distinguir con mucho cuidado entre paquetes de entrada y de salida, así evitar de forma real los paquetes falsificados.

Como primer paso debe determinar el paradigma de permisos, esto consiste en elegir una de las dos posturas para la política de seguridad, la más sencilla y segura ya desde el inicio, consiste en negar cualquier conexión que no tenga su permiso explícito. Esta postura permitirá controlar el tráfico con una lista de transacciones o tareas que sí permite y esta lista será sin duda más pequeña y comprensible.

¿Qué hace el router con los paquetes?

Una vez que el Firewall ha terminado la inspección de un paquete, tiene dos alternativas respecto del análisis:

- Aceptar el paquete como cualquier otro *router*.
- Rechazar el paquete.

Sería deseable, que el *router* tenga un registro de las acciones tomadas, especialmente si el paquete es desechado, saber que se intentó y porqué no se permitió.

Devolución de códigos de error icmp.

Cuando se desecha un paquete, el *router* generalmente enviará un mensaje ICMP al origen informándole de las causas del problema, se deberá plantear permitir esto y que nivel de información se permitirá que envíe el *router*, pues es una fuente de riesgos.

Hay varios elementos que deberemos considerar:

- ¿Qué mensaje debemos devolver?
- ¿Puede afrontar los gastos de generar y devolver los códigos de error?.
- Devolver esos códigos, ¿Permitirá al atacante obtener demasiada información sobre el filtrado de paquetes?

La siguiente tabla nos ayudará en la elección de qué errores permitir

Tipo numérico	Nombre simbólico	Descripción
0	echo-reply	Una respuesta de ping
3	destination-unreachable	Un mensaje de estado de error general; un enrutador a lo largo de la trayectoria hasta el destino es incapaz de entregar el paquete al siguiente destino; lo usa traceroute.
4	source-quench	Flujo de nivel de red IP entre dos enrutadores, o entre un enrutador y un *host*.
5	Redirect	Un mensaje de enrutamiento que se devuelve al remitente cuando un enrutador determina que existe una trayectoria más corta.
8	echo-request	Una petición de ping
11	time-exceeded	Un mensaje de enrutamiento que se devuelve cuando el contador de saltos máximos de un paquete (TTL) se sobrepasa; lo usa *traceroute*.
12	parameter-problem	Aparecen valores inesperados en el encabezado del paquete IP.

Tabla 6 Tipos habituales de mensajes ICMP

En resumen, en el momento de usar y configurar el filtrado de paquetes debemos recordar que:

- Es muy recomendable editar las reglas para filtrado con el sistema fuera de línea respecto a Internet.
- Debe guardar los archivos de filtros por duplicado y con comentarios.

- Es mejor (más serio) usar siempre direcciones IP y no nombres de anfitrión.
- Al hacer cambios, hay que restaurar las reglas desde el inicio, para evitar malas interacciones entre nuevas y viejas.

Filtrado por dirección.

La más simple, aunque no más común de las formas de filtrado es el filtrado por dirección, permite restringir el flujo de paquetes basándose en la dirección fuente y/o destino, sin tener en cuenta que protocolos están involucrados. Pero recuerde que las direcciones incluidas en la cabecera pueden ser falsificadas, así pues, a menos que usemos algún tipo de autentificación criptográfica entre nosotros y la máquina exterior en concreto, no sabrá nunca si se comunica con él o alguien que finge serlo.

Filtrado por puerto fuente.

La mayor utilidad del método anterior es la detección de paquetes falsos y evitar conexiones desde lugares claramente inseguros, sin embargo, la política de seguridad, comúnmente incluirá unas reglas de un grano más grueso (más generalistas, para simplificar y obtener una seguridad mayor) las cuales detallarán que servicios pueden establecerse y cuales no. Para poder realizar esta selección y dado que los paquetes TCP/IP o UDP/IP no contienen ningún campo con información sobre la aplicación de nivel superior que los ha creado, deberemos guiarnos por el puerto destino u origen.

Así, como si sabe cuales son los puertos "bien conocidos" que esperan para ofrecer servicios podrá utilizarlos para negar el acceso a ellos desde su red y/o evitar el uso de los mismos desde fuera hacia adentro.

No deja de ser arriesgado tomar decisiones para el filtrado basadas en el puerto fuente, ya que el problema de base radica en que solamente se puede confiar en un puerto fuente tanto como se confié en la máquina de donde viene.

Debe restringir al máximo posible los números de puerto locales, sin importar que su número sea muy bajo.

Como muchos servicios usan puertos al azar sobre el 1023 para sus clientes, y puesto que algunos servicios usan puertos superiores al 1023, con frecuencia deberá aceptar paquetes de entrada que vengan de servidores que no sean dignos de confianza.

¿Cómo elegir un router-firewall con filtrado de paquetes?

Deberá tener un funcionamiento de filtrado suficientemente bueno para sus necesidades, aunque de hecho, en la mayoría de los Firewalls de Internet el factor crítico respecto de limitaciones no es el filtrado sino la conexión de la red.

El filtrado de paquetes es una operación que depende directamente del tamaño por paquete, de ahí que mientras menor sea el tamaño de estos, mayor cantidad de ellos podrán ser revisados en una unidad de tiempo.

Puede ser un router dedicado o una pc

Si dispone de muchas redes o múltiples protocolos, probablemente necesitará un *router* dedicado. El *routing* de paquetes por computadora por lo común no tiene la velocidad y/o flexibilidad de los *routers* dedicados y podrá encontrar que necesitará máquinas demasiado grandes y caras.

Si esta filtrando un sólo enlace con Internet, tal vez sólo necesite enrutar los paquetes entre dos Ethernet, lo cual está dentro de las capacidades de un 486, y esta máquina bien puede ser más barata que un router dedicado.

Cualquiera que sea el mecanismo, servir de Firewall es todo lo que este debe hacer, mientras más complejo sea el *router* y su configuración tanto más fácil será que se equivoque o pase por alto algo que no debería.

El filtrado además causa un significativo impacto sobre la velocidad de enrutado, de manera que puede afectar al buen funcionamiento de las redes internas y el tiempo de respuesta para las peticiones de clientes externos.

Debe permitir una especificación simple de regla, ya que el filtrado es complicado para comenzar, y lo es más por los detalles y peculiaridades de los diversos protocolos.

Las reglas deben aplicarse en el orden especificado, pues al reordenarse las reglas, se dificulta saber lo que pasa y como interactuarán entre sí.

Si hay algunas peculiaridades o errores al momento de fusionar o reordenar los conjuntos de reglas, se vuelve imposible descubrir lo que hará el sistema con un determinado grupo de filtros.

Lo más importante es que reordenar las reglas puede deshacer un conjunto que sólo trabajaría bien si no hubiera sido reordenado.

Debe ser posible aplicar las reglas por separado a los paquetes de entrada y salida, basándonos en la tarjeta de red (interfaz).

Para mayor comodidad, flexibilidad y capacidad de funcionamiento, deberá especificar un conjunto de reglas distinto para los paquetes entrantes y salientes en cada interfaz.

Una limitante que desafortunadamente comparten muchos sistemas para filtrado de paquetes es que les permiten examinar los paquetes sólo mientras salen del sistema.

Esta limitante trae consigo tres problemas:

- El sistema siempre esta "fuera" de sus propios filtros.
- Es difícil o imposible detectar paquetes falsificados.
- Es muy difícil configurar tales sistemas si tiene más de dos interfaces.

El primer problema será si el *router* deja ver sólo los paquetes de salida, los paquetes que se dirigen al propio *router* nunca están sujetos al filtrado de paquetes; como resultado tendrá que el filtrado no incluye al *router* en sí.

Como segundo problema tiene que si el Firewall sólo puede filtrar los paquetes salientes, es difícil o imposible detectar paquetes falsos provenientes del exterior.

El tercer problema de filtrar sólo la salida es que puede ser complicado configurar el filtrado en el Firewall cuando éste tiene más de dos interfaces.

Debe ser capaz de registrar los paquetes aceptados y/o los rechazados.

Como mínimo deberá asegurar que registra en un *log* los paquetes que rechaza y la causa. El modo más sencillo de aprender es ver cómo y cuándo han atacado de forma fallida nuestra red. Asimismo son prueba evidente de la existencia de un atacante.

Esto es el filtrado de paquetes con lo que se espera que se entienda parte de lo que se ocupará en la construcción del Firewall.

CAPÍTULO

3

Diseño e implementación del Firewall

3. Diseño e implementación del Firewall

Con base en la información obtenida hasta este momento se puede tomar la decisión sobre el diseño del Firewall, así como los requerimientos que deberá tener la instalación, los servicios que dará el Departamento de Administración de Servidores a través de Internet y la configuración de estos en el Firewall.

La Dirección General de Servicios de Cómputo Académico (DGSCA) de la UNAM es la entidad universitaria encargada de la operación de los sistemas centrales de cómputo académico y de las telecomunicaciones de la UNAM; su esfuerzo más amplio es la capacitación en tecnología de la información, de prospección e innovación y de asimilación de estas tecnologías en beneficio de la Universidad y de la sociedad en general.

Fig. 8. Organigrama DTD.

Dirección de Telecomunicaciones Digitales (DTD)

Subdirección de Operación
Equipos de Respuesta Rápida
Departamento de Administración de Servidores
Departamento de Tarificación y Servicios al Público
Coordinación de Control de Ingresos y Egresos
Departamento de Relación con Proveedores y Clientes
Departamento de Ventas
Centro de Atención
Departamento de Suministro de Potencia

La DTD se originó en 1989 con el fin de crear la Red Integral de Telecomunicaciones de la UNAM. Las subdirecciones y coordinaciones de la DTD trabajan conjuntamente para dar una mejor optimización de los recursos y servicios que ofrece al público en general, así como a la comunidad universitaria, sus instalaciones se encuentran en la DGSCA.

Subdirección de redes

La Subdirección de redes opera, coordina y mantiene la Red Universitaria de Datos (RedUNAM) a nivel técnico y administrativo. Además analiza y dirige los lineamientos a seguir en la evolución de la misma Red y se encuentra integrada por:

- Proyectos Especiales y Atención a Usuarios
- Operación de la Red
- Administración de Servidores

Departamento de Administración de Servidores

El Departamento de Administración de Servidores se creó en 1997 como parte de la descentralización de la Coordinación de Servicios de Red (CSR). Perteneciendo desde esa fecha a la Subdirección de Redes.

El Departamento de Administración de Servidores Centrales de RedUNAM, está dedicado a mantener los sitios principales de la Universidad Nacional Autónoma de México, dependencias externas tales como: el periódico "La Jornada", "CENAPRED", entre otros, así como también las principales bases de datos de la UNAM, tales como la Dirección General de Administración Escolar. Mantiene los principales servicios de *web hosting* y correo, los servidores de correo principales de la UNAM son servidor.unam.mx, www.correo.unam.mx, actualmente se tienen un total de usuarios de correo en servidor.unam.mx de 22045 y en correo.unam.mx es de 17765, cerca de 500 sitios *web* alojados en todos los servidores a cargo de este Departamento

Actualmente se han hecho reestructuraciones en la operación y administración de los servidores, por los actuales administradores, migración de los servicios de correo, migración de algunos servidores y servicios a otras máquinas y Sistemas Operativos, investigación de otros servicios, atención diaria a los usuarios de dichos servicios, etc.

Dichos administradores se encuentran laborando en el edificio de la DGSCA, en un espacio recién modificado y acondicionado en espacios para trabajo de los administradores y becarios, en dicho espacio se encuentran las máquinas de los administradores y algunas otras de propósitos generales, este espacio es el que está involucrado en la propuesta para la implementación del Firewall.

Considerando la importancia de los servicios que éste Departamento ofrece es necesaria la implementación de una herramienta como un Firewall para ofrecer seguridad a dichos servicios.

3.1 ¿Comprar o diseñar el Firewall?

El comprar o diseñar un Firewall no es una decisión fácil, puesto que se deben tomar en cuenta las características que nos ofrecen las dos posiciones, comprar o bien diseñar, además de responder a la siguiente pregunta ¿Cuánto puede ofrecer una organización por su seguridad?. Un simple paquete de filtrado Firewall puede tener un costo mínimo debido a que la organización necesita un ruteador conectado al Internet, y dicho paquete ya está incluido como estándar del equipo. Un sistema comercial de Firewall provee un incremento más a la seguridad pero su costo puede ser muy elevado dependiendo de la complejidad y el número de sistemas protegidos. Si la organización cuenta con gente capacitada, un Firewall también puede ser construido con software de dominio público pero este ahorro de recursos repercuten en términos del tiempo de desarrollo y el despliegue del sistema Firewall. Finalmente requiere de soporte continuo para la administración, mantenimiento general, actualización de software, reparación de seguridad, e incidentes de manejo.

Para ayudar a la decisión se muestra en la siguiente tabla un resumen de las características de las dos posturas (comprar o diseñar).

DISEÑAR	COMPRAR
- Bajo costo en componentes de diseño.	-Los Firewalls comerciales son costosos.
- Se obtiene un Firewall justo a sus necesidades.	- La documentación sobre como configurarlo es en ocasiones de acceso restringido.
- Construido con software de dominio público.	- Se tiene soporte por parte de la compañía..
- Requiere de tiempo para el desarrollo y la puesta en marcha del Firewall.	- Entra en funcionamiento en cuanto se lleva a cabo su compra.
- Requiere de soporte continúo para la administración, monitoreo y actualización del software.	- Incremento en la seguridad.

Tabla 7. Comprar o diseñar un Firewall.

En este punto la decisión de colocar un Firewall para el Departamento no es difícil ya que como se ha mencionado, la seguridad es indispensable para cualquier sitio que se encuentre conectado a Internet lo que no será sencillo es decidir ¿que arquitectura de Firewall usar?, además en qué sistema operativo estará y qué Firewall será.

En el caso del Departamento se tienen máquinas que se usan para administrar remotamente servidores lo cual da un panorama general de la inseguridad que se tiene al conectarse a sistemas en producción desde máquinas que están abiertas a Internet lo cual da un pase directo a los servidores cuando estas máquinas son comprometidas.

Existen tres decisiones básicas en el diseño o la configuración de un Firewall:

- La primera de ellas, la más importante, hace referencia a la política de seguridad de la organización propietaria del Firewall. La configuración y el nivel de seguridad potencial será distinto en una empresa que utilice un Firewall para bloquear todo el tráfico externo hacia el dominio de su propiedad (excepto, quizá, las consultas a su página *web*) frente a otra donde sólo se intente evitar que los usuarios internos pierdan el tiempo en la red, bloqueando por ejemplo todos los servicios de salida al exterior excepto el correo electrónico. Sobre esta decisión influyen, aparte de motivos de seguridad, motivos administrativos de cada organismo.
- La segunda decisión de diseño a tener en cuenta es el nivel de monitorización, redundancia y control deseado en la organización; una vez definida la política a seguir, hay que definir como implementarla en el Firewall indicando básicamente que se va a permitir y que se va a denegar. Para esto existen dos aproximaciones generales: o bien se adopta una postura restrictiva (denegamos todo lo que explícitamente no se permita) o bien una permisiva (permitimos todo excepto lo explícitamente negado); evidentemente es la primera la más recomendable de cara a la seguridad, pero no siempre es aplicable debido a factores no técnicos sino humanos (esto es, los usuarios y sus protestas por no poder ejecutar tal o cual aplicación a través del Firewall).

•Finalmente, la tercera decisión a la hora de instalar un sistema de Firewall es meramente económica: en función del valor estimado de lo que desee proteger, debe gastar más o menos dinero, o no gastar. Un Firewall puede no entrañar gastos extras para la organización, o suponer un desembolso de varios millones: seguramente un departamento o laboratorio con pocos equipos en su interior puede utilizar una PC con Linux, Solaris o FreeBSD a modo de Firewall, sin gastarse nada en él, pero esta aproximación evidentemente no funciona cuando el sistema a proteger es una red de tamaño considerable; en este caso se pueden utilizar sistemas propietarios, que suelen ser caros, o aprovechar los *routers* de salida de la red, algo más barato pero que requiere más tiempo de configuración que los Firewall sobre UNIX en PC (en el apéndice A se listan algunos Firewalls).

Estas decisiones, aunque concernientes al diseño, son básicamente políticas; la primera decisión técnica a la que nos vamos a enfrentar a la hora de instalar un Firewall es elemental, ¿donde se situara para que cumpla eficientemente su cometido?, si aprovechamos como Firewall un equipo ya existente en la red, por ejemplo un *router*, no tenemos muchas posibilidades de elección: con toda seguridad hemos de dejarlo donde ya está; si por el contrario utilizamos una máquina UNIX con un Firewall implementado en ella, tenemos varias posibilidades para situarla respecto a la red externa y a la interna. Sin importar donde situemos al sistema hemos de recordar siempre que los equipos que queden fuera del Firewall, en la zona de riesgo, serán igual de vulnerables que antes de instalar el Firewall; por eso es posible que si por obligación hemos tenido que instalar un Firewall en un punto que no protege completamente nuestra red, pensemos en añadir Firewalls internos dentro de la misma, aumentando así la seguridad de las partes más importantes.

Mantener sencillo el Firewall, de manera que sea más fácil asegurarlo, debe proporcionar el conjunto más pequeño de servicios con los menores privilegios posibles para cumplir con sus tareas.

Con la información recabada podremos hacer un análisis con base en nuestras necesidades y recursos.

Se debe tener diseñado y probado un plan de contingencia para casos en los que el *bastión* resulta atacado con éxito. Sólo anticipando lo peor y preparando planes para ello se tendrá la posibilidad de impedirlo. Por lo que es importante hacer el análisis de riesgos como se mencionó en el punto 1.4.

Identificación de:

Recursos:

Tangibles, máquinas SUN sparc, PC's con Linux, impresoras, unidades de cintas de respaldo. Las máquinas se utilizan para la administración remota de los servidores en producción (de correo, *web hosting*, respaldos, entre otros).

Intangibles, las cuentas que se encuentran en cada máquina las de los administradores principalmente, las contraseñas, la información almacenada, la confianza de cada máquina para las demás, y la reputación del Departamento.

Amenazas:

Las amenazas a las que se encuentra expuesto el Departamento se clasifican con base en los recursos tangibles y no tangibles.

Para los tangibles puede haber robo de las máquinas si no se tiene la suficiente seguridad (que en el caso del Departamento se podría asegurar que no sucedería, porque el mecanismo que se tiene no permite pasar a personas ajenas al trabajo y de ser así debe ser con un registro previo). Otra amenaza de este tipo podría ser un terremoto, pero es poco probable que esto destruyera nuestros recursos, ya que el Departamento se encuentra en una zona segura para esto. También pueden haber

errores humanos al haber malas conexiones en los equipos. Obviamente de estas amenazas nuestro Firewall no podría salvarnos, pero es importante que se tomen en cuenta y se hace algo por evitarlas.

Del tipo de amenazas que el Firewall si puede proteger son de las intangibles, el robo de contraseñas, usurpación de identidad, robo de información, etc. Esto se lograra a través de tener una buena configuración del Firewall.

Decisión de comprar o diseñar:

La decisión de comprar o diseñar el Firewall es de la organización, como ya se había mencionado y tomando los tres principios antes mencionados. El primero hace referencia a la política de seguridad de la organización, el segundo a la política de acceso ya sea con una postura restrictiva, o bien, una permisiva, la tercera se refiere a la económica y está en función del valor estimado de lo que se desea proteger.

Con base en las necesidades y recursos con los que se cuente, el diseño puede ser desde un sólo equipo que funcione como Firewall, hasta un diseño con varios componentes, lo cual garantiza una fuerte protección. Se puede escoger de las arquitecturas mencionadas en el capítulo 2, sea cual sea la arquitectura a elegir se deben tomar otras medidas para seguridad ya que un Firewall no basta para asegurar toda la protección del sitio.

Por lo tanto, en el Departamento se decidió diseñar un "Firewall de arquitectura de anfitrión con doble acceso con enrutamiento interno". En un sistema operativo Linux, el Firewall está integrado en el *Kernel*, es un software libre, esta decisión se tomó con base en los tres principios básicos para elegir el Firewall:

•Qué tanta seguridad queremos dar a nuestras máquinas. Un sólo componente que se encargue de toda la seguridad es riesgoso debido a que si un punto deja de funcionar o es atacado todas las demás máquinas se verán afectadas.

•Respecto al segundo principio se tomó la postura "denegamos todo lo que explícitamente no se permita".

•La tercera política fue la que más influyó en la decisión para elegir este Firewall, la cantidad de dinero que se desea invertir en la implementación del Firewall, ya que no se cuenta con muchos recursos y sabemos que un Firewall Linux sobre una PC es funcional para una red como la nuestra.

El hecho que se haya elegido un Firewall que teóricamente no sea el más adecuado no indica que en el Departamento y en la práctica no sea suficiente para dar una seguridad aceptable, con la ayuda de otros mecanismos de seguridad como los son herramientas en cada una de las máquinas de la red que se intenta proteger para que en caso de haber intrusiones al Firewall las máquinas no estén desarmadas del todo, tales herramientas adicionales las podemos verificar en el apéndice B.

Pero esto no indica que no se pueda implementar cualquier otro Firewall en el Departamento pero esto dependerá de lo que se desee en un futuro invertir, después de que se hayan realizado pruebas con el Firewall que se propone.

3.2 ¿ Qué máquina utilizar?

Una vez seleccionado el Firewall deseado, en este caso un "Firewall Anfitrión de doble acceso con enrutamiento interno", debemos escoger los componentes que nos servirán para el fin deseado:

Un paso importante es elegir el tipo de máquina a utilizar, evaluando la elección de:

•Software

•Hardware

3.2.1 Elección de Software

La elección del software se refiere principalmente a la elección del sistema operativo que se instalará y obviamente con el que se trabajará, además de la velocidad con la que trabajará la máquina.

Sistema operativo

.

Debe ser un sistema operativo con el que se esté familiarizado, pues se terminará personalizando tanto la máquina como el sistema operativo extensamente.

Se necesita una máquina fiable que ofrezca, la gama de servicios Internet que deseé proporcionar a los usuarios.

Unix es el sistema operativo más popular para ofrecer servicios de Internet, y las herramientas están ampliamente disponibles para construir anfitriones *bastión* en sistemas UNIX. Si ya se tiene máquinas UNIX, se debe evaluar UNIX para el anfitrión *bastión*. Se recomienda que se pruebe con UNIX por las ventajas ya mencionadas.

Si se elige UNIX, se debe pensar que versión y distribución, por regla general, si la versión de UNIX que se seleccione tiene asociado un grupo de usuarios, es posible que sea lo suficientemente conocida para poder confiar en ella.

Por lo anterior, el sistema operativo que se eligió es Linux Red Hat, una versión de UNIX, software libre al menos hasta la versión 9.x y con documentación en la red lo cual da una gran facilidad de manejo, además, que en el Departamento se maneja principalmente el sistema operativo UNIX, lo cual facilitará la administración del Firewall.

Rapidez

La máquina no necesita ser rápida; de hecho es mejor que no sea muy potente, el por qué radica en que si es comprometida debe ser lo menos útil para el atacante, ya que por el contrario si se le proporciona una máquina potente la usará en nuestra contra, también evitaremos que los usuarios la utilicen para otros propósitos que no corresponden a esta máquina. Existen varias buenas razones, además del costo, para hacer el anfitrión *bastión* tan potente sólo lo necesario a fin de que cumpla con su trabajo, pero no más. No se necesitan muchos caballos de fuerza para proporcionar los servicios requeridos del anfitrión *bastión*. El *bastión* no tiene mucho que hacer, en realidad estaremos más bien limitados por la velocidad de conexión que por la del CPU.

Existen varias razones para no exagerar el tamaño del *bastión*:

- Porqué una máquina lenta es un blanco menos tentador.
- Si está comprometida, una máquina lenta es menos útil.
- También es menos atractiva para que la comprometan usuarios internos, pues de lo contrario podría acabar siendo utilizada para trabajos que no le corresponden.

3.2.2 Elección del hardware.

Se requiere una configuración de hardware confiable, así que se debe seleccionar una máquina base y dispositivos que no sean lo más nuevo en el mercado pero tampoco seleccionar algo tan viejo que no se puedan encontrar refacciones, elegiremos un punto medio.

Ya hemos dicho que no se necesita un CPU muy potente, pero si será necesario un uso intensivo de memoria RAM y discos duros para *swap*. Asimismo no serán necesarias capacidades gráficas importantes. Por ejemplo, el servidor *web* apache, se ejecuta y controla perfectamente en modo consola desde un terminal de UNIX.

3.2.3 Ubicación del Firewall

El anfitrión *bastión* debe estar en una ubicación que sea segura físicamente, por algunas razones:

- Es importante asegurar adecuadamente una máquina contra un atacante que tiene acceso físico a ella; hay demasiadas formas de que el atacante pueda comprometerla.
- El anfitrión *bastión* proporciona gran parte de la funcionalidad real de su conexión a Internet, y si se pierde, daña o roba, su sitio podría realmente desconectarse. Con toda certeza, se perderá acceso a algunos de los servicios.
- El anfitrión *bastión* debe estar en una habitación cerrada, con aire acondicionado y ventilación adecuados.

Por lo tanto, el Firewall del Departamento estará en el *site* del Departamento, el cual está cerrado y sólo se ingresa con tarjeta y clave, el aire acondicionado es el adecuado por lo que es el mejor lugar para colocarlo.

3.3 Servicios Seleccionados

Además de los servicios que se ofrecerá al mundo exterior, se puede proporcionar cualquier servicio que necesite la organización para tener acceso a Internet, o que desee ofrecer a está. No ofertar ningún servicio que no se vaya a utilizar. (La Tabla A1 del apéndice B muestra un conjunto de servicios y su descripción, para tener una mejor referencia de los servicios que se pueden ofertar).

Se dividirán los servicios en cuatro clases:

- Seguros: se proporcionan mediante filtrado de paquetes.
- Inseguros debido a como se proporcionan pero que pueden asegurarse: serán ofertados solo en el *bastión*.

- Inseguros debido a cómo se proporcionan, pero que no pueden asegurarse: deberán desactivarse y ser instalados solo si son realmente necesarios.
- Aquellos no utilizados: deben ser desactivados.

Los servicios que se darán en el Departamento se decidieron basándose en la seguridad que proporcionan y la necesidad de tenerlos aunque no sean muy seguros, las descripciones de los servicios en el capitulo 1 en el punto 3 nos ayudarán a decidir mejor sobre qué servicios permitir en nuestra red y con ayuda de la tabla A1, la cual nos da una descripción y recomendación para algunos servicios de red.

Correo electrónico

El servicio de correo electrónico es como ya se había mencionado inseguro, pero en el caso de la red que se protegerá no es indispensable, ya que los administradores tienen otras cuentas de correo en un servidor publico, por lo que no es necesario habilitar el correo en cada uno de las máquinas, por lo que no se habilitara este servicio.

Transferencia de datos (FTP)

Este servicio es uno de los más inseguros, pero en ocasiones se abrirá para la transferencia de archivos por lo que se permitirá, pero se tendrá mucho cuidado de su configuración.

Telnet

Este es un servicio inseguro pero como FTP en ocasiones necesario, se permitirá solo Telnet de salida pero no se podrán hacer conexiones de Telnet a nuestra

red esto con el fin de evitar que intrusos quieran estar espiando la información que se transmite por Telnet y que como ya se ha mencionado viaja sin cifrar por la red, y al permitir que se realicen conexiones de Telnet hacia fuera sólo se harán transferencias a sitios en los cuales confiemos.

Secure Shell

Este servicio como ya se mencionó es uno de los más seguros debido al cifrado que realiza en la transferencia de datos, es el que más se utiliza en las conexiones a los servidores en producción por lo que se permitirá este servicio de salida y entrada solo a algunos equipos.

WWW

Este servicio también es inseguro pero necesario por lo cual se permitirá su uso, las medidas que se tomarán es que sólo se permitirán conexiones a sitios relativamente seguros, y se filtrarán los sitios que no son seguros con reglas como las mostradas en el apéndice C.

Servicios de Información

Este servicio además de ser poco usado en la actualidad es inseguro y no necesario para el Departamento, por lo que no se implementará.

Servicios de conferencia en tiempo real

Este servicio como se señaló en el punto 1.3.6, es inseguro además que en el departamento no es necesario por lo que no se implementará.

Servicio de nombres

El Servicio de Nombres de Dominio (*DNS o Domain Name Service*) permite que cada sitio tenga información sobre sus propios anfitriones y pueda encontrar la información para otros sitios. DNS soporta SMTP, FTP, Telnet y casi cualquier otro servicio que necesiten los usuarios, quienes quieren escribir " telnet sitio.com" en lugar de "telnet 10.20.230.2". Este servicio sí es necesario por lo que se permitirá en el Firewall.

Servicio de administración de redes

Las dos herramientas para administración de redes más comunes son *ping* y *traceroute*. Ambas se conocen como herramientas UNIX, por ser los primeros en utilizarlas, pero ahora están disponibles de alguna forma en casi todas las plataformas en Internet. No tienen sus propios protocolos, ocupan el mismo protocolo fundamental, el Protocolo de Control de Mensajes de Internet (ICMP o Internet *Control Message Protocol*), estos servicios se implementarán ya que son indispensables para monitorear si nuestra red está funcionando correctamente, se filtraran sólo permitiendo algunos errores ICMP, según la tabla 6 del capítulo 2.

Sistema de Archivos de red

Los servicios de archivos de red NFS, NIS, AFS, son inseguros y obsoletos en algunos casos, no se permitirán la salida ni entrada de éstos, en caso de requerirse se hará la petición justificando la necesidad de estos servicios y se tomará la decisión con base en las políticas que se tienen en el Departamento, las cuales podemos ver en el Apéndice C .

Sistema de ventanas

Este servicio es cómo ya se había mencionado, unos de los más inseguros por lo que no se permitirán las conexiones a través de este servicio desde Internet.

Sistemas de impresión

Este servicio es necesario para los administradores, pero si el servidor de impresión está en la misma red no es necesario abrirlo para máquinas que estén fuera de ella, ya que las máquinas internas podrán utilizar la impresora sin necesidad de pasar por el Firewall.

Muchos servicios incluyen vulnerabilidades que los atacantes pueden explotar desde afuera y comprometer el Firewall, por lo que debemos desactivar cualquier cosa que no utilicemos y seleccionar lo que utilizaremos con mucho cuidado.

Otro punto importante es que no permitiremos cuentas de usuario en el Firewall, solo la del administrador del sistema (root) y una de para el staff, con la cual se administrará el Firewall, hay varias razones, como las siguientes:

- Vulnerabilidad de las mismas cuentas.
- Vulnerabilidad de los servicios requeridos para soportar las cuentas.
- Reducida estabilidad y confiabilidad de la máquina.
- Alteración inadvertida de la seguridad del Firewall por los usuarios.
- Incremento en la dificultad para detectar ataques.

Si se necesitara en un futuro habilitar algunas cuentas éstas las tendremos vigiladas revisándolas con cuidado y verificándolas con regularidad.

3.4 Construcción y configuración del Firewall

En esta parte ya sabemos qué servicios dará el Firewall, el siguiente paso es construirlo, y seguiremos los siguientes pasos:

1.- Asegurar la máquina.
2.- Deshabilitar todos los servicios no requeridos.
3.- Instalar o modificar los servicios que se proporcionarán
4.- Reconfigurar la máquina de modo apropiado
5.- Revisión de seguridad
6.- Conectar la máquina a la red en la que se utilizará

Asegurar la máquina

Se instaló lo menos posible en la máquina en cuanto a software se refiere, ya que es más sencillo instalar elementos que borrarlos completamente después. Ya que el sistema esté funcionando, no será difícil agregar componentes si se requieren. Instalaremos la máquina en un sitio seguro y con un sistema operativo estándar fiable.
Se tomó en cuenta la instalación de parches y las recomendaciones de seguridad al construir el Firewall, pero debemos estar al día en vulnerabilidades o noticias que puedan ser útiles en el mantenimiento.

Un punto importante es el respaldo de la contabilidad del sistema, la cual arroja el demonio de Syslog, en un archivo especial para los registros del Firewall, para esto se edito el /etc/syslog.conf indicando que se guardará en /var/log/fwlog, agregando la siguiente línea:

```
kern.info                    /var/log/fwlog
```

El respaldo se hace semanalmente junto con los demás servidores en cintas magnéticas, las cuales se guardan en un sitio seguro bajo llave.

Es recomendable que se tengan éstos respaldos porque es con lo que se podrá comparar que el Firewall no ha sido comprometido o monitorear los intentos de ataque.

Deshabilitar todos los servicios no requeridos.

Existen servicios esenciales para el funcionamiento como*: init, swap, syslogd, inet, cron*, etc., éstos no pueden ser deshabilitados, para conocer los que debe deshabilitar seguirá esta pauta: si no lo necesita o no sabe lo que hace, entonces lo desactiva.

Algunos ejemplos de servicios a eliminar son:
nfsd, biod, autmount, mountd, rexd, etc..

Para deshabilitar los servicios en Solaris basta con comentar en el archivo /etc/inetd.conf el servicio, se coloca un signo # al inicio de la línea. En Linux se quitan del /etc/xinetd.conf y en los scripts relacionados.

Instalar los servicios que deseemos proporcionar.

Se decidió que servicios ofrecer, cómo ya se mencionó en secciones anteriores, se debe proteger estos servicios y mantenerlos vigilados, esto se hará con ayuda de TCP Wrapper y *xinetd*, los cuales se describen en el apéndice B. Estos se instalaron ya que como hemos mencionado anteriormente no basta con poner un sólo mecanismo de seguridad, si no que se debe instalar y configurar otras herramientas que integraran en realidad el Firewall, ya que por sí solo Iptables no podría denominarse como Firewall si no se integra con otros mecanismos de seguridad.

Reconfigurar la máquina de modo apropiado.

Esto significa ultimarla para el proceso de explotación, reconfigurando el *kernel* si es necesario, quitando programas innecesarios y dejando tantos sistemas de archivos como podamos en modo lectura (únicamente).
Además, las herramientas necesarias podrían ser eliminadas al acabar la tarea, siendo por tanto obligatoria su reinstalación en caso de necesidad, pero aumentando el nivel de seguridad.

Revisión de seguridad

Hacer una revisión de seguridad y establecer el modo normal de operación. Esto lo hicimos con algunas herramientas para saber cuál será el comportamiento normal, utilizando herramientas como las siguientes:

para buscar archivos que tengan encendido el *bit seguid* o *setgid*.
root@fw # find / -type f \ (-perm –04000 –o –perm –02000 \) -ls

para escanear que puertos encendidos en máquinas remotas así como la local
root@fw # nmap localhost

de forma local
root@fw # lsof -i | grep LISTEN

Estos procedimientos nos arrojan información, donde se toman decisiones de los puertos a cerrar y si se encuentran archivos *setuid*, se quitan dichos permisos si no se desea que lo tengan, hay más herramientas que sirven para hacer revisiones (en el apéndice B se describen algunas).

Conectar la máquina a la red.

Se conectó la máquina ya asegurada y se activó el Firewall como se muestra en el apéndice C, ahora por ningún motivo se debe conectar a red antes de protegerla, ya que como experiencia se noto que en cuanto una máquina es detectada los atacantes entran en acción y la atacan, por lo que si no se es cuidadoso la máquina puede estar comprometida mucho antes de ponerse en funcionamiento

Configuración de la Máquina

Se siguieron los pasos anteriores para configurar el Firewall en una máquina con Linux Red Hat versión 8. Anteriormente se habló del filtrado de paquetes y ahora se explicará cómo se realizó en el Firewall.

Con Linux, el filtrado de paquetes está incorporado en el *kernel*. La implementación exacta del filtro depende de la versión que se está ejecutando. El filtrado de paquetes se introdujo por primera vez en la versión del *kernel* 1.3.X y las reglas de los filtros habían sido configuradas y manipuladas por un programa desarrollado por Jos Vos: *ipfwadm*.

Cuando se publicó el *kernel* 2.1.102, muchos se sorprendieron de ver que sus antiguos filtros de paquetes ya no funcionaban. Con el lanzamiento del *kernel* 2.1.103, el cambio estaba documentado. Ya estaba disponible un nuevo programa, llamado ipchains, que funcionaba con el nuevo código de filtrado de paquetes. En la actualidad existe iptables a partir del *kernel* versión 2.4 de Linux, este filtro es el que utilizaremos, dando aquí ejemplos de cómo utilizarlo (para más detalles remitirse al apéndice A). Como siempre, lo mejor es ir a la documentación que acompaña a los programas si se quiere un completo repaso de sus características.

Activación del reenvió de IP

Una vez que el *kernel* está en su sitio y el sistema está ya funcionando, se necesita activar el reenvío de IP, éste viene desactivado de forma predeterminada. Para activar esta opción, se debe editar el archivo /etc/sysconfig/network y cambiar FORWARD_IPV4 de *false* a *true*. Esto activará el reenvío de IP cada vez que se arranque el sistema, si no es así sólo haga:

```
root@fw # echo 1 > /proc/sys/net/ipv4/ip_forward
```

Nota: tome en cuenta que si hace lo de la línea anterior en línea de comando, cada que reinicie la máquina lo tendrá que hacer.

A continuación, hay que colocar en su sitio las reglas del cortafuegos. A medida que vaya creciendo el sistema, los filtros tendrán que ser definidos en el momento oportuno. El mejor lugar para empezar el Firewall es justo antes de haber configurado los dispositivos de red usando ifconfig. En Red Hat Linux, las interfaces están configuradas en el *script* /etc/rc.d/init.d/network. Este *script* tiende a ser uno de los primeros ejecutados en los niveles 2,3,5. En estos directorios de nivel de ejecución el enlace simbólico para el programa de *shell* /etc/rc.d/init.d/network, es S10network. Un posible método para configurar las reglas de filtrado de paquetes correctamente antes de que se configuren las interfaces, es crear un *script* de *shell* como /etc/rc.d/init.d/firewall, despues es necesario, crear los enlaces en los directorios del nivel de ejecución con el programa de *shell* de cortafuegos con lo siguiente:

```
/usr/bin/sh
##firewall.sh ##
for i in rc{2,3,5}.d ; do
cd /etc/rc.d/$I
ln –s .. /init.d/firewalls S5firewalls
cd ..
done
```

Este minibucle de *shell* cambia el directorio a cada uno de los niveles de ejecución rc2.d, rc3.d y rc5.d y crea un enlace con el programa de *shell* de cortafuegos en /etc/rc.d/init.d/ que se ejecutará antes de que la red se vuelva a recuperar.

Un Firewall de filtrado de paquetes

Se eligió el filtro de paquetes Iptables por lo que se recordará algo sobre lo que es un Firewall de filtrado de paquetes.

El Firewall de filtrado de paquetes consta de una lista de reglas de aceptación y denegación. Estas reglas definen explícitamente los paquetes que se permiten pasar y los que no a través de la interfaz de red. Las reglas del Firewall usan los campos del encabezado del paquete, para decidir si enrutar un paquete hacia su destino, eliminar el paquete o bloquear un paquete y devolver una condición de error a la máquina emisora.

Estas reglas se basan en la tarjeta de interfaz de red especifica y en la dirección IP del *host*, las direcciones IP origen y destino del nivel de red, los puertos de servicio UDP y TCP de la capa de transporte, los indicadores de conexión TCP, los tipos de mensaje ICMP de nivel de red y en si el paquete es entrante o saliente.

Un Firewall de filtrado de paquetes funciona en las capas de red y transporte, como se muestra en la Figura 6 del capítulo 2.

En el capítulo 2 se mencionó que en el filtrado de paquetes hay dos posturas:

- "No todo lo específicamente permitido está prohibido" (Negación preestablecida).
- "No todo lo específicamente prohibido está permitido" (Permiso preestablecido).

Se eligió la postura de "negación preestablecida", por ser la mas segura asumiendo que lo que no se conoce puede dañarnos, aunque para los usuarios esto no sea lo más grato.

En esta parte se mostrarán algunos ejemplos de la configuración para el Firewall. En el apéndice C se colocan las reglas utilizadas en el Firewall, es importante señalar que éstas son solamente ejemplos y que cada quien debe diseñar sus reglas de acuerdo a sus necesidades.

Configuración de red en las máquinas

Cuando ya se ha configurado la máquina de tal manera que no corra riesgo alguno de que los servicios dentro de ella sean activados o usados por intrusos, entonces podemos proceder a configurar la red para las máquinas.
Se darán configuraciones para Solaris y Linux.

Configuración	Solaris	Linux
El nombre del *host*	Editar el archivo /etc/hostname.<tarjeta>	Editar el archivo /etc/sysconfig/network-scripts/ifcfg-<tarjeta>
Nombre de los *hosts* conocidos	Editar el archivo /etc/hosts	Editar el archivo /etc/hosts
La máscara de red	Editar el archivo /etc/netmasks	Editar el archivo /etc/sysconfig/network-scripts/ifcfg-<nombre tarjeta>
El *router*	Editar el archivo /etc/defaultrouter	/etc/sysconfig/network
Configuración de DNS	Editar el archivo /etc/resolv.conf	Editar el archivo /etc/resolv.conf

Tabla 8. Configuraciones de red

Ejemplo para una máquina Solaris:

defaultrouter
172.16.0.28

netmasks
anteriormente tenía
#132.248.120.0 255.255.255.0
Se coloca la nueva red y su máscara
172.0.0.0 255.0.0.0

hosts
Internet *host* table
127.0.0.1 localhost
172.16.0.29 hafnio loghost
Las dos interfaces del Firewall
172.16.0.28 fw.servidores.unam.mx
132.248.120.98 fw.servidores.unam.mx

resolv.conf
nameserver 172.16.0.28
nameserver 132.248.120.98
nameserver 132.248.204.1
nameserver 132.248.10.2

El DNS para las máquinas será el mismo Firewall pero es importante colocar los DNS externos para que funcione.
hostname
fw

Después de editar los archivos se debe reiniciar la máquina para que se realicen los cambios.

Puede hacer los cambios desde línea de comandos con el comando *ifconfig*, pero al reiniciar la máquina se perderá la configuración, ya que tomará los datos que se encuentren en estos archivos, por lo que es recomendable editar los archivos.

Puede corroborar que los datos estén correctos con:

root@fw # netstat -rn

Destination	Gateway	Flags	Ref	Use	Interface
172.0.0.0	172.16.0.29	U	1	0	hme0
224.0.0.0	172.16.0.29	U	1	0	hme0
default	172.16.0.28	UG	1	0	
127.0.0.1	127.0.0.1	UH	61	7707	lo0

en las máquinas Linux seguir los mismos pasos que se mencionan en la tabla anterior y para cargar los cambios basta con hacer lo siguiente:

```
root@fw # /etc/init.d/network restart
```

Es recomendable hacer pruebas antes de comenzar a cerrar los servicios para saber si existe comunicación entre las máquinas, con el fin de asegurar que cuando cerremos los servicios las máquinas funcionen correctamente y no sea un problema sólo de comunicación.

Los componentes del Firewall fueron:

Hardware

- PC 486
- Disco duro de 4GB
- Memoria de 32MB
- Unidad de CDROM
- Floppy de 3.5
- Mouse, teclado
- 2 tarjetas de red

Software

- Sistema operativo Linux
  - Iptables
  - Nmap, lsof, ssh.

El esquema que se tendría con el Firewall es el siguiente:

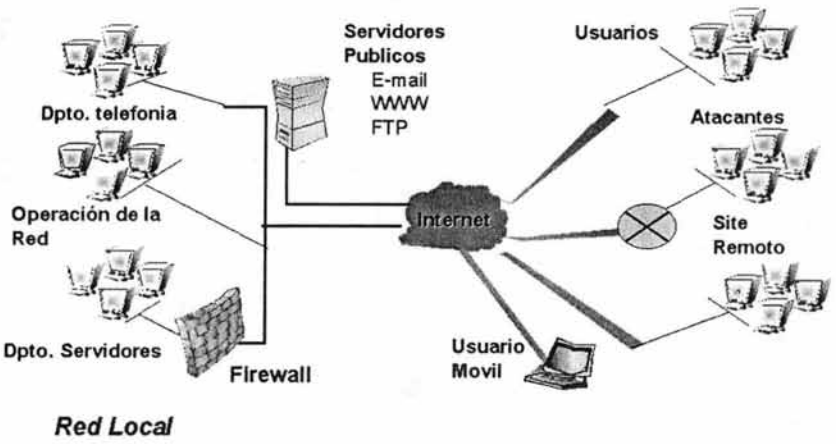


Fig. 9.Implementación del Firewall en la red.

3.5 Monitoreo y Depuración del Firewall

Si se realiza un buen trabajo al diseñar un Firewall que se adecue a sus necesidades, debe ser muy sencillo mantenerlo. Las tareas de mantenimiento caen en tres categorías:

- Mantenimiento
- Monitoreo del sistema
- Mantenerse actualizado

Una vez que se diseñó el Firewall, el mantenerlo no debe ser una tarea que requiera de un gran esfuerzo, debido a que gran parte del trabajo de mantenimiento puede automatizarse.

El mantenimiento es el eterno ciclo de pequeñas tareas que debe hacerse para mantener el Firewall a salvo. Hay 3 principales tareas que debe hacer cotidianamente:

•Respaldar el Firewall
•Administrar las cuentas
•Administrar el espacio en el disco

Respaldar el Firewall

Debe asegurarse de respaldar todas las partes del Firewall, es recomendable hacer respaldos incrementales y completos, con el fin de tener la información para cuando se necesite, esta tarea la podemos automatizar colocando los pasos para respaldar el sistema en un *script* el cual puede ser ejecutado desde un *cron* el cual se ejecutará cada determinada fecha según se prefiera.

Administración de cuentas

La administración de cuentas (agregar o quitar cuentas, cambiar contraseñas, etc.) es una tarea muy importante, esta tarea también pude ser automatizada creando una programa, como ya se había dicho las cuentas en el Firewall no son recomendables, pero si no hay manera de evitarlas es importante tenerlas bien configuradas.

Administración del espacio en disco

La información siempre se extiende hasta llenar el espacio disponible en el disco aún en máquinas que casi no tienen usuarios. Los usuarios dejan cosas en rincones extraños del sistema, "en forma provisional", y luego se quedan ahí, lo cual ocasiona problemas, por ejemplo, no sabrá si algún archivo usted lo puso alguna vez o es un regalo de algún intruso, por lo que si se tiene el sistema bien vigilado y no se colocan archivos en lugares incorrectos se ahorra trabajo.

Desafortunadamente no hay un modo automático para hallar la basura, los usuarios (en particular los administradores de sistemas) que pueden escribir en

cualquier parte del disco, son demasiado impredecibles. Para esto alguna persona deberá revisarlo periódicamente.

Monitoreo del sistema

Un aspecto importante del mantenimiento de los Firewalls es el monitoreo del sistema el cual sirve para indicar varias cosas:

- ¿Está comprometido el Firewall?
- ¿Qué clase de ataques han intentado contra él?
- ¿Funciona adecuadamente?
- ¿Puede el Firewall proporcionar el servicio que los usuarios necesitan?

Para llevar a cabo esta tarea debe tener algunas herramientas de monitoreo. Algún analizador de red (sniffer), además de saber que observar, como que su sistema no se llene y revisar las bitácoras, las cuales deben registrar los siguientes casos:

- Todos los paquetes rechazados, conexiones negadas e intentos frustrados
- Por lo menos la hora, el protocolo y nombre del usuario de cada conexión exitosa hacia o a través del Firewall
- Todos los mensajes de error del Firewall

Esto servirá para tomar las medidas necesarias para un buen funcionamiento del Firewall.

Una parte importante del mantenimiento del Firewall es el mantener actualizado el sistema y uno mismo como administrador se debe estar al pendiente de los avances en este campo, ya que todos los días ocurren cosas nuevas: se descubren y explotan nuevos errores, ataques nuevos y la disponibilidad de nuevas herramientas.

Para mantenerse actualizado puede hacerse en principio, una suscripción en listas de correo interesantes que manejen este tipo de trabajos, en grupos de trabajo o en foros de discusión que se pueden encontrar en la red.

Además de mantener el Firewall también debe estar preparado para responder a los incidentes de seguridad con los siguientes pasos[13].

Manejo de incidentes

- Regla número 1: No asustarse
- Regla número 2: Documentar todo
- Regla número 3: Planeación

    Planeación

        Paso 1: Identificar y entender el problema

        Paso 2: Contener o detener el problema

        Paso 3: Confirme el diagnóstico y termine el daño

        Paso 4: Restaure el sistema

        Paso 5: Ataque el problema

        Paso 6: Llame al seguro (en caso de que tenga uno)

- Detección

        Revisión de los archivos de bitácora

        Reparando los daños

- Reportar

Uno de los pasos más importantes es el de reportar; desgraciadamente por alguna razón se da parte de lo ocurrido, debe hacerse notar que esto no es para ponerse al descubierto ni poner en tela de juicio si se es o no buen administrador, si no es para contribuir a que a otros administradores no les suceda lo mismo o para que sepan cómo fue solucionado el problema y tomar sus precauciones.

[13] Pasos sugeridos por el Departamento de Seguridad de Computo de la DGSCA, publicado en un documento titulado "Seguridad en Unix I"

Existen sitios en donde se pueden reportar los incidentes de seguridad:

- CERT (Computer Emergency Response Team) en www.cert.org
- UNAM-CERT en www.seguridad.unam.mx, www.unam-cert.unam.mx

Configurar un Firewall no es tarea fácil, requiere un tiempo considerable de reflexiones previas y de planificación. Dependiendo del número de servicios que se ofrezcan, las reglas del Firewall en iptables pueden ser desde simples y cortas hasta sumamente complejas. Los Firewall pueden ser filtros de paquetes que ofrezcan sólo servicios a través de ellos, o pueden involucrar servidores *proxy*, que actúan en representación de un cliente solicitando un servicio. No importa cómo se implemente un Firewall, proporciona una de las mejores herramientas de seguridad que se pueden usar para proteger la red, reiterando una vez más que esto debe ser acompañado con otras herramientas de seguridad.

3.6 Ejemplos y pruebas

Se comenzará con las reglas del Firewall, pero antes es recomendable que lea el apéndice A donde encontrará documentación de iptables para que pueda entender las reglas y los componentes que ofrece ya que en está parte se dará de manera general lo que hace cada opción de iptables.

Nota: Como ya se había mencionado iptables viene integrado en el *kernel* por lo que no se tuvo que instalar, si tiene una *kernel* antes del 2.4 debe instalar **iptables** antes de hacer esto.

Ejemplos:

Para ver cuál es el estado de nuestras 3 cadenas:

```
root@fw # iptables –L
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Si la salida obtenida no se parece a lo anterior, significa que no se tienen instalados los módulos correspondientes en el *kernel*, o que no hay un Firewall funcionando. Por el contrario si la salida es igual, quiere decir que el Firewall está aceptando todos los paquetes, o lo que es lo mismo, no está filtrando nada ya que las 3 cadenas tienen como política predeterminada "ACCEPT".

Nota: Para entender las cadenas que maneja iptables revise el apéndice A.

En este momento si damos un ping a *localhost* y saldrá lo siguiente:

```
root@fw # ping localhost
PING localhost.localdomain(127.0.0.1) from 127.0.0.1:56(84) bytes of data
64 bytes from 132.248.120.108: icmp_seq=0. time=0. ms
64 bytes from 132.248.120.108: icmp_seq=1. time=0. ms
.....
.....
root@fw # iptables –A INPUT –p icmp –s 127.0.0.1 -j DROP
```

Esto significa: Agregar (-A) la siguiente regla a la cadena de entrada (INPUT) al recibir un paquete del tipo icmp (-p) con origen (-s) 127.0.0.0 y con cualquier destino (ya que no se especifico), enviar ese paquete (-j) a DROP (desecharlo).
Para revisar la regla que se introdujo:

```
root@fw # iptables –L

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  127.0.0.0            0.0.0.0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

En la cadena de entrada (INPUT) se agregó una regla, la cual evitará que se haga *ping* desde nuestra propia máquina, pero esto no es muy útil cuando queremos que otros no nos hagan *ping* entonces se hará la siguiente regla, pero antes de agregar la que impida que nos hagan *ping* se borra la que impide que nosotros lo hagamos:

```
root@fw # iptables -D INPUT -p icmp -source 127.0.0.0 -j DROP
```

Con (-D) borramos la regla, pero si no queremos teclear nuevamente toda la regla podemos hacerlo así :

```
root@fw # iptables -D INPUT 1
```

regla para impedir que hagan ping desde cualquier origen a cualquier destino (-d 0/0)

```
root@fw # iptables -A INPUT -p icmp -s 0/0 -d 0/0 –j DROP
```

Si alguien intenta dar un ping a la máquina Firewall (fw.servidores.unam.mx)

```
maq@externa # ping fw.servidores.unam.mx
PING localhost.localdomain(127.0.0.1) from 127.0.0.1:56(84) bytes of data
```

Se quedará esperando una respuesta que nunca llegará

Más ejemplos:

Para la tabla de entrada (INPUT)

Para permitir consultas DNS

```
root@fw # iptables -I INPUT -p udp –s0/0 -d 0/0 --dport 53 -j ACCEPT
root@fw # iptables -I INPUT -p tcp -s 0/0 -d 0/0 --dport 53 -j ACCEPT
```

Para permitir conexiones por *Secure Shell*, solo de ciertas IP´s (administradores remotos....), recuerde que el puerto asociado a ssh es el 22. La red que se permitirá es la 132.248.120.96/255.255.255.240.

```
root@fw #  iptables -A INPUT -p tcp -i eth0 -s 132.248.120.96/255.255.255.240  --dport 22 -j ACCEPT

root@fw #  iptables -A INPUT -p udp -i eth0 -s 132.248.120.96/255.255.255.240  --dport 22 -j ACCEPT
```

Si se permitiera ftp, solo para la red 132.248.120.96/28

```
root@fw #  iptables -A INPUT -p tcp -i eth0 -s 132.248.120.96/28--dport 20 -j ACCEPT
root@fw #  iptables -A INPUT -p tcp -i eth0  -s 132.248.120.96/28  --dport 21 -j ACCEPT
```

Cualquier conexión a puertos entre el 0 y el 1023 no permitirla después de permitir los servicios anteriores(SSH, FTP, DNS)

```
root@fw #  iptables -A INPUT -p tcp -s 0/0 -d 0/0 --dport 0:1023 -j DROP
```

Para la tabla de salida (OUTPUT)

Para permitir consultas DNS

```
root@fw #  iptables -I OUTPUT -p udp -s 0/0 -d 0/0 --dport 53 -j ACCEPT
root@fw #  iptables -I OUTPUT -p tcp -s 0/0 -d 0/0 --dport 53 -j ACCEPT
```

Permitir a cualquiera que esté en nuestra red utilizar SSH a cualquier parte

```
root@fw #  iptables -A OUTPUT -p tcp --dport 22 -s 0/0 -d 0/0 -j ACCEPT
root@fw #  iptables -A OUTPUT -p udp --dport 22 -s 0/0 -d 0/0 -j ACCEPT
```

Cualquier otro servicio que no sea el anterior negarlo

```
root@fw #  iptables -A OUTPUT -p tcp -s 0/0 -d 0/0 --dport 0:1023 -j REJECT
```

Las reglas mostradas anteriormente son algunas reglas sencillas para introducirse un poco a Iptables, con referencia a las cadenas INPUT y OUTPUT, existe también la de FORWARD, en el apéndice C podrá encontrar más reglas que se probaron, pero éstas sólo deben ser tomadas como ejemplo y crear sus propias reglas de acuerdo a sus necesidades.

CAPÍTULO

4

Políticas de seguridad en el Departamento

4. Políticas de Seguridad en el Departamento

La palabra política asusta a mucha gente, ya que hace referencia a documentos impenetrables creados por desconocidos, dichos documentos son ignorados por la mayoría de las personas. Las políticas de seguridad y la importancia que tienen para el buen funcionamiento del Departamento son muy importantes, por lo que se hablará de lo que se debe tener en cuenta cuando se desarrollen, así como la importancia que tienen al intentar proteger el sitio y la relación que guardan con las reglas que se implementaran en el Firewall.

Se mostrará en el apéndice C las políticas que se tienen en general y se detallan las que nos servirán para la implementación del Firewall.

4.1 Justificación de las políticas de seguridad

La mayoría de los administradores consideran que es deseable una política de seguridad, pero también creen por experiencia que intentar conformar una puede ser bastante complicado. Conformar una política de seguridad es mucho más entretenido que hacer frente a los efectos colaterales de no tener una. A la larga, se pasará menos tiempo en reuniones discutiendo sobre seguridad si se soluciona cuanto antes.

Política:

Una política de seguridad es un conjunto de reglas y prácticas que regulan la manera como se maneja, protege y distribuye información sensible a través de la organización. Es el marco que sirve de apoyo para contar con un sistema confiable. Una política de seguridad se enuncia normalmente en

términos de sujetos y objetos. Un sujeto es algo activo en el sistema; usuarios, procesos y programas. Un objeto es algo que está sujeto a la acción de un objeto; archivos, directorios, dispositivos, etc.[14].

La política de seguridad es (debería ser) un documento que está (debería estar) firmado por la alta gerencia de la empresa y mediante el cual se especifican distintos aspectos referentes a la seguridad informática de la empresa. Estos aspectos pueden ser desde cuantas letras han de tener las contraseñas de los usuarios corporativos y cada cuanto tiempo han de cambiarlas, qué protocolos van a permitir que hablen las máquinas internas con las externas y en su caso quien va a poder iniciar la conexión, y hasta la política que se va a seguir para permitir el acceso restringido a recursos internos.

¿Qué hace la política?

- La política detalla las actividades que están, o no, permitidas.
- Los pasos a seguir para obtener la protección adecuada así como que hacer en caso de presentarse un incidente de seguridad.
- Establece responsabilidades y derechos.
- Una política de seguridad explica las sanciones que se impondrán a los que infrinjan una regla.

4.2 ¿Cómo conformar las políticas de seguridad?

La política de seguridad es una forma de comunicarse con los usuarios. Debe decirles lo que deben de saber para tomar las decisiones que deben tomar respecto a la seguridad.

[14]ZARAGOZA, Fernando, "Seguridad en Unix 1", http://newman.posgrado.unam.mx/ds/cursos/seguridad , sección "Políticas de seguridad"

Es importante que la política sea explícita y comprensible del por qué deben tomarse ciertas decisiones. Casi nadie acata las instrucciones a menos que entienda por qué son importantes. Una política que especifique lo que debe hacerse, pero no por qué, está destinada al fracaso. Tan pronto como la gente que la escribió se vaya u olvide el por qué tomó la decisión, dejará de tener efecto.

El rol de las políticas

Las políticas juegan tres tipos de roles. El primero: aclarar que es lo que se requiere proteger y porqué. Segundo, establecer de manera clara la responsabilidad para lograr esa protección. Tercera, proveer la base sobre la que se interpreta y/o resuelve cualquier conflicto posterior. Lo que una política **NO** debe hacer es listar amenazas específicas, máquinas e individuos por nombre. Las políticas deben ser generales y actualizarse conforme se necesite.

Consejos generales para la elaboración de políticas

- Las políticas deben ser documentos sencillos y entendibles.
- Antes de empezar a desarrollar las políticas de una organización es conveniente determinar cuál es el método que se seguirá:
  - Lo que no está expresamente permitido está prohibido
  - Lo que no está expresamente prohibido está permitido
- Las políticas deben ser apoyadas por la dirección.
- Las políticas deben ser concisas.
- Deben contener un balance entre protección y productividad.
- Actualizarse regularmente para que reflejen la evolución de la organización.
- Las políticas deben ser un esfuerzo en conjunto de la organización y no un esfuerzo de una sola persona.
- Todos los afectados por las políticas deben tener la oportunidad de revisarlas antes de ser emitidas.
- Deben reflejar derechos, obligaciones, así como sanciones en caso de presentarse.
- Se definen los derechos y responsabilidades de los administradores del sistema.

4.3 Clasificación de las políticas

No existe una clasificación de políticas general, ésta es definida por el tipo de organización y por los administradores con base en las necesidades de cada organización. La que se presenta a continuación se tomó con base en la experiencia que se tiene sobre los servicios del Departamento de Administración de Servidores, DGSCA. Cada administrador debe hacer un análisis de la política adecuada para su organización.

4.3.1 Políticas de control de acceso

Una de las partes más importantes en las políticas de seguridad son las secciones que describen quién tiene acceso al sistema y quién puede dar acceso a otro usuario al sistema.

La primera pregunta que una política de seguridad debe responder es: ¿a quién se le está permitido acceder a los sistemas?. Por ejemplo a personas que no son parte del grupo de administradores no se les permitirá usar terminales, y las que sí son integrantes del grupo de administradores se les permitirá, con una cuenta previa en la cual, como se había mencionado anteriormente se tendrá mucho cuidado con todas las cuentas que existan y aún más las que tienen privilegios.

4.3.2 Políticas sobre contraseñas

Normalmente las contraseñas se consideran la primera línea de defensa contra el acceso no autorizado, es importante que las políticas de seguridad cubran este aspecto. Existen tres elementos para una buena política sobre contraseñas.

Primero, los usuarios deberán ser instruidos sobre la manera de seleccionar contraseñas seguras. Esta instrucción deberá tomar la forma de una lista de puntos a seguir cuando se seleccionen contraseñas.
Segundo, es importante hacer notar a los usuarios la importancia de mantener sus contraseñas en secreto. Las contraseñas jamás deberán ser escritas en papel o en agendas, calendarios, etc.. Almacenar las contraseñas en línea es una mala idea.
Tercero, a los usuarios no se les permite compartir su contraseña, y por ende su cuenta, con otras personas.
Una última sugerencia es incluir las instrucciones para el cambio de contraseñas, esto podría motivar a los usuarios que leyeron las políticas a ignorarlas debido a que no saben cómo cambiar su contraseña.

4.3.3 Políticas sobre uso de hardware y software

Las políticas sobre el uso de hardware y software en el Departamento son importantes ya que se debe saber que usuarios están autorizados para usar el equipo y de qué forma además de especificar que software está permitido y el manejo de éste.

Las licencias de software y los derechos de autor han tomado gran importancia en los últimos años. Es importante explicar que algún programa o todos los utilizados en el sistema se encuentran bajo licencia o protegidos por derechos de autor (nacionales o internacionales en su caso), y no se pueden violar los términos de éste convenio. La violación de las licencias de software y derechos de autor es una ofensa seria, a menudo sancionadas con grandes multas u otras sanciones.

Hay que decidir la manera de configurar las computadoras para conseguir la cantidad de seguridad conveniente para el sistema. El tener sistemas similares configurados de manera similar ayudará inicialmente, y será de utilidad para detectar una configuración que no es igual a las otras.

Debe fijarse un calendario para la auditoria de los sistemas de seguridad. En él debe anotarse la ejecución del software de auditoria del sistema de archivos, software de auditoria de la red y software de auditoria de la configuración. Generalmente se usa *cron* para ejecutar estos paquetes y los registros de salida son escaneados o enviados por correo electrónico al administrador diariamente.

También se debe comprobar la aparición de parches del fabricante para el software que ejecuta en el sistema. Deben aplicarse esas actualizaciones tan pronto como se tiene conocimiento de ellas para estar prevenidos contra los ataques más recientes.

Es importante hacer notar que bajo ciertas circunstancias será necesario que el administrador del sistema examine los archivos privados de algún usuario, como parte de las actividades cotidianas del administrador o cuando se investigue un incidente de seguridad. La política de seguridad deberá explicar esto a los usuarios, para evitar situaciones desagradables con los mismos.

Con esto no se ésta sugiriendo el acceso irrestricto por parte del administrador del sistema y que pueda hacer lo que quiera, sin embargo, es apropiado informar a los usuarios de las acciones que pueden tomarse por el administrador del sistema si ocurren determinados eventos.

Escribir una política de seguridad no es suficiente. El paso más importante es hacer que cada usuario del sistema esté enterado de dicha política. La manera más fácil de hacer que cada usuario del sistema se entere de las políticas es entregarles una copia de las mismas al momento de recibir su cuenta en el sistema. Para asegurarse de que lea y entienda dichas políticas se le pedirá que firme una copia y se guardará en el centro de cómputo.

Si no puede lograr a pesar de su mejor esfuerzo una política de seguridad, la respuesta más segura es: documente lo que está haciendo, por qué, cuáles son las políticas

existentes, que intenta hacer y por qué piensa que la situación es mala, esto para avalar que usted hizo algo.

4.4 Las políticas y las reglas del Firewall

Se ha definido cómo se deben realizar las políticas de seguridad, es importante señalar que las políticas son importantes en la realización de las reglas de filtrado en el Firewall y que guardan una estrecha relación en especial las que tienen que ver con control de acceso a los sistemas. Por ejemplo, si una de nuestras reglas sobre acceso al Firewall es que sólo los administradores pueden hacerlo, pensemos que sólo existe una máquina desde la cual nos podremos conectar y dicha máquina tiene la IP 132.248.120.98, y sólo se permite el acceso por medio de ssh, la regla de filtrado con iptables sería :

```
root@fw #  iptables -A INPUT -p tcp -i eth0 -s 132.248.120.96/255.255.255.240  --dport 22 -j ACCEPT
root@fw #  iptables -A INPUT -p udp -i eth0 -s 132.248.120.96/255.255.255.240  --dport 22 -j ACCEPT
```

Cualquier otro servicio que no sea el anterior negarlo

```
root@fw # iptables -A OUTPUT -p tcp -s 0/0 -d 0/0 --dport 0:1023 -j REJECT
```

Como ya se ha mencionado tener políticas de seguridad es importante puesto que nos serán de gran ayuda para la elaboración de las reglas de filtrado siendo más fácil hacer estas reglas sabiendo que se permitirá y que no.

En el apéndice C se encuentran las políticas de seguridad así como las reglas de filtrado para el Firewall. Algunas de ellas con respecto al control de acceso y uso de software nos servirán para hacer filtrado de paquetes las demás son el complemento del que se había hablado para un plan de seguridad aceptable.

CONCLUSIONES

Las organizaciones que están conectadas o que desean conectarse a Internet para utilizar los servicios que éste nos proporciona deben estar concientes de la necesidad de tener algún tipo de protección para disminuir el riesgo que existe al hacerlo en cuanto a seguridad se refiere.

El implementar un Firewall en conjunto con herramientas de seguridad correctamente configuradas es una buena alternativa para lograr cierta protección y seguridad, aclarando que un Firewall al igual que cualquier otro esquema de seguridad no garantizan un 100 % seguridad.

El nivel de seguridad que se desea implementar depende del valor que la organización de a su información, entendiendo al Firewall como una inversión y no como un gasto.

Para elegir adecuadamente el tipo de Firewall a implementar es necesario realizar un análisis formal que nos permita valorar nuestras necesidades, beneficios y requerimientos que éste necesita.

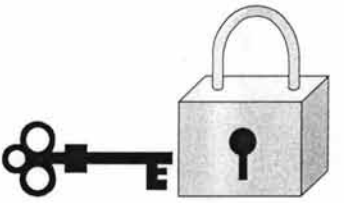
Los Firewalls comerciales son costosos y la documentación sobre cómo configurarlos es en ocasiones de acceso restringido. Linux es una alternativa para implementar seguridad incluyendo un Firewall a bajo costo, además que la configuración de un Firewall y muchas otras herramientas en equipos Linux se facilita gracias a la disponibilidad de información en Internet.

Cualquier esquema de seguridad necesita procedimientos formales que indiquen la forma en la que se debe administrar y utilizar las herramientas, que es el caso del

Firewall, sin procedimientos no se puede asegurar el uso adecuado de ninguna herramienta.

Con el desarrollo de este trabajo la implementación de un Firewall en el Departamento de Administración de Servidores, DGSCA, es muy útil para la protección de la seguridad que en él se maneja, un Firewall Linux es una buena opción.

APÉNDICES



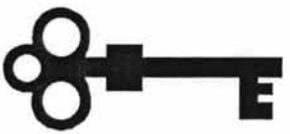
Esta parte consta de los siguientes apéndices:

El apéndice A. Descripción de Iptables, lista de algunos Firewalls

El apéndice B. Herramientas de seguridad, tabla de algunos servicios de red UNIX.

El apéndice C. Políticas y reglas del Firewall, contiene políticas del Departamento de Administración de Servidores, DGSCA, y las reglas de filtrado del Firewall.

APÉNDICE A



Iptables y Firewalls.

IPTABLES

Linux contiene utilidades avanzadas para *filtrado de paquetes*, el proceso de controlar los paquetes de red cuando entran, se mueven o salen de su sistema dentro del *kernel*. Los *kernels* anteriores al 2.4 trabajaban con *ipchains* para efectuar el filtrado de paquetes y usaban listas de reglas que se aplicaban a los paquetes en cada paso del proceso de filtrado. La presentación del kernel 2.4 trajo consigo Iptables (también llamado *netfilter*), que es parecido a *ipchains* pero con mejoras en el funcionamiento y en el control disponible a la hora de filtrar paquetes.

Aquí nos centraremos en las bases del filtrado esencial de paquetes, se define las diferencias entre ipchains e iptables, se explica las diferentes opciones disponibles con comandos iptables, y se muestra cómo las reglas de filtrado se pueden conservar tras el reinicio del sistema.

Nota importante

El mecanismo predeterminado del Firewall en la versión 2.4 del kernel puede usar el comando iptables, pero no se puede usar si ya se está ejecutando ipchains. Si ipchains está presente durante el arranque, el kernel avisará que hay un error y no podrá arrancar iptables.
Estos errores no afectan la funcionalidad del comando ipchains.

El *kernel* de Linux contiene la característica interna de filtrado de paquetes, que le permite aceptar algunos de ellos en el sistema mientras que intercepta y para a otros. El filtro de red kernel 2.4 tiene tres *tablas* internas o *listas de reglas, s*on las siguientes:

- FILTRO — esta es la tabla por defecto para manejar paquetes de red.
- NAT — esta tabla se usa para alterar paquetes que crean una nueva conexión.
- MANGLE — esta tabla se usa en tipos específicos de alteración de paquetes.

Cada una de estas tablas tiene un grupo de *cadenas* internas que corresponden a las acciones llevadas a cabo por el filtro de red en el paquete.

Las cadenas internas para la tabla filtro son las siguientes:

- *INPUT* — Esta cadena sirve sólo para paquetes recibidos por medio de una interfaz de red.
- *OUTPUT* — Esta cadena sirve para paquetes enviados por medio de la misma interfaz de red que recibió los paquetes.
- *FORWARD* — Esta cadena sirve para paquetes recibidos en una interfaz de red y enviados en otra.

Las cadenas internas para la tabla nat son las siguientes:

- *PREROUTING* — Esta cadena altera paquetes recibidos por medio de una interfaz de red cuando llegan.
- *OUTPUT* — Esta cadena altera paquetes generados localmente antes de que sean dirigidos por medio de una interfaz de red.
- *POSTROUTING* — Esta cadena altera paquetes antes de que sean enviados por medio de una interfaz de red.

Las cadenas internas para la tabla mangle son las siguientes:

- *PREROUTING* — Esta cadena altera paquetes recibidos por medio de una interfaz de red antes de que sean dirigidos.
- *OUTPUT* — Esta cadena altera paquetes generados localmente antes de que sean dirigidos por medio de una interfaz de red.

Cada paquete de red recibido o enviado de un sistema Linux está sujeto a al menos una tabla.

Un paquete puede que sea verificado contra muchas, muchas reglas dentro de la lista de reglas antes de llegar al final de una cadena. La estructura y propósito de estas reglas puede variar, pero normalmente buscan identificar un paquete que viene de o se

dirige a una dirección IP en particular o un conjunto de direcciones al usar un determinado protocolo y servicio de red.
Independientemente de su destino, cuando un paquete cumple una regla en particular en una de las tablas, se asignan a un *objetivo (target)* particular, o una acción a aplicárseles. Si la regla especifica un objetivo ACCEPT para un paquete que la cumpla, el paquete se salta el resto de las verificaciones de la regla y se permite que continúe hacia su destino. Si una regla especifica un objetivo DROP, el paquete "se deja caer", significando esto que no se permite que el paquete acceda al sistema y no se envía ninguna respuesta de vuelta al servidor que envió el paquete. Si una regla especifica un objetivo REJECT, el paquete se deja caer, pero se envía un mensaje de error al emisor.

Cada cadena tiene una política ACCEPT, DROP, o REJECT sobre el paquete, o si no, puede enviarlo al espacio de usuario con QUEUE. Si ninguna de las reglas de la cadena se aplican al paquete, entonces el paquete se trata de acuerdo a la política por defecto de las cadenas.
El comando iptables le permite configurar estas listas de reglas, así como configurar nuevas cadenas y tablas para ser usadas en situación particular.

Diferencias entre iptables e ipchains

En un primer momento, ipchains e iptables parecen ser bastante similares. Ambos métodos de filtrado de paquetes usan cadenas o reglas que operan con el kernel de Linux para decidir no solo qué paquetes se permite entrar o salir, sino también qué hacer con los paquetes que cumplen determinadas reglas. Sin embargo, iptables ofrece un método mucho más extensible de filtrado de paquetes, proporcionando al administrador un nivel de control mucho más refinado sin tener que aumentar la complejidad del sistema entero.
Más concretamente, los usuarios que se encuentren cómodos con ipchains deberían tener cuidado con las siguientes diferencias significativas entre ipchains e iptables antes de utilizar iptables:

- *Bajo iptables, cada paquete filtrado se procesa únicamente usando las reglas de una cadena, en lugar de hacerse con múltiples.* Por ejemplo, un paquete FORWARD que llega al sistema usando ipchains tendrá que pasar por las cadenas INPUT, FORWARD, y OUTPUT para llegar a su destino. Sin embargo iptables, solo envía paquetes a la cadena INPUT si su destino es el sistema local y tan solo los envía a la cadena OUTPUT si el sistema local es quien genera los paquetes. Por esta razón, deberá estar seguro de situar la regla destinada a interceptar un paquete en particular en la cadena adecuada que será la que vea el paquete.
- La principal ventaja es que tendrá un control más refinado sobre la disposición de cada paquete. Si está intentando bloquear el acceso a un sitio web en particular, ahora es posible bloquear los intentos de acceso desde clientes que están en máquinas que utilicen nuestro servidor como pasarela (gateway). Una regla OUTPUT que deniegue el acceso no prevendrá más el acceso a las máquinas que utilicen nuestro servidor como pasarela.
- *El objetivo DENY ha sido cambiado por DROP.* En ipchains, los paquetes que cumplían una regla en una cadena podían ser redirigidos a un objetivo DENY, que dejaba caer el paquete en silencio. Este objetivo deberá cambiarse a DROP con iptables para obtener el mismo resultado.
- *El orden es importante al poner opciones en una regla de una cadena.* Anteriormente, con ipchains, no era muy importante cómo se ordenasen las opciones de las reglas a la hora de escribirla. El comando iptables es un poco más reticente sobre el lugar que ocupan las diferentes opciones. Por ejemplo, ahora deberemos especificar el puerto origen y destino después del protocolo (ICMP, TCP, or UDP) que vayamos a utilizar en una regla de una cadena.
- *Cuando especificamos las interfaces de red que vamos a usar en una regla, deberemos utilizar solo interfaces de entrada (opción -i) con cadenas INPUT o FORWARD y las de salida (opción -o) con cadenas FORWARD o OUTPUT.* Esto es necesario debido al hecho de que las cadenas OUTPUT no se utilizan más con las interfaces de entrada, y las cadenas INPUT no son vistas por los paquetes que se mueven hacia las interfaces de salida.

Esto no es una lista muy extensa de cambios, dado que iptables es fundamentalmente un filtro de red completamente reescrito.

Opciones usadas en comandos iptables

Las reglas que permiten a los paquetes ser filtrados por el kernel se ponen en ejecución ejecutando el comando iptables. Cuando use el comando iptables, debe especificar las opciones siguientes:

- *Packet Type* — dicta qué tipo de paquetes filtra el comando.
- *Packet Source or Destination* — dicta qué paquetes filtra el comando basándose en el origen o destino del paquete.
- *Target* — dicta qué acción se lleva a cabo en paquetes que cumplen los criterios mencionados anteriormente.

Las opciones usadas con la regla dada iptables deben Estar agrupadas lógicamente, basándose en el propósito y en las condiciones de la regla general, para que la regla sea válida.

Tablas

Un aspecto muy potente de iptables es que se pueden utilizar múltiples tablas para decidir el destino de un paquete en particular, dependiendo del tipo de paquete que se esté monitorizando y de qué es lo que se va a hacer con el paquete. Gracias a la naturaleza extensible de iptables se pueden crear tablas especializadas que se almacenarán en el directorio /etc/modules/<*kernel-version*>/kernel/net/ipv4/netfilter para objetivos específicos. Piense que iptables es capaz de ejecutar múltiples conjuntos de reglas ipchains en las cadenas definidas, en las que cada conjunto cumple un rol específico.

La tabla por defecto, llamada filter, contiene las cadenas estándar por defecto para INPUT, OUTPUT, y FORWARD. Esto es parecido a las cadenas estándar que se utilizan con ipchains. Además, por defecto, iptables también incluye dos tablas adicionales que realizan tareas de filtrado específico de paquetes. La tabla nat se puede utilizar para modificar las direcciones de origen y destino grabadas en un paquetes, y la tabla mangle permite alterar los paquetes de forma especializada.
Cada tabla contiene cadenas por defecto que realizan las tareas necesarias basándose en el objetivo de la tabla, pero se pueden configurar fácilmente nuevas cadenas en el resto de las tablas.

Estructura

Muchos comandos iptables tienen la siguiente estructura:

```
iptables [-t <table-name>] <command> <chain-name> <parameter-1> \
    <option-1> <parameter-n> <option-n>
```

En este ejemplo, la opción *<table-name>* permite al usuario seleccionar una tabla diferente de la tabla filter por defecto que se usa con el comando. La opción *<command>* es el centro del comando, dictando cuál es la acción específica a realizar, como pueda ser añadir o borrar una regla de una cadena particular, que es lo que se especifica en la opción *<chain-name>* . Tras *<chain-name>* se encuentran los pares de parámetros y opciones que realmente definen la forma en la que la regla funcionará y qué pasará cuando un paquete cumpla una regla.
Cuando miramos a la estructura de un comando iptables, es importante recordar que, al contrario que la mayoría de los comandos, la longitud y complejidad de un comando iptables puede cambiar en función de su propósito. Un comando simple para borrar una regla de una cadena puede ser muy corto, mientras que un comando diseñado para filtrar paquetes de una subred particular usando un conjunto de parámetros específicos y opciones puede ser mucho más largo. Al crear comandos iptables puede ser de

ayuda reconocer que algunos parámetros y opciones pueden crear la necesidad de utilizar otros parámetros y opciones para especificar algo de los requisitos de la opción anterior. Para construir una regla válida, esto deberá continuar hasta que todos los parámetros y opciones que requieran otro conjunto de opciones hayan sido satisfechos. Si teclea "iptables –h" para ver una lista detallada de la estructura de los comandos iptables.

Comandos

Los comandos le dicen a iptables que realice una tarea específica. Solamente un comando se permite por cada cadena de comandos iptables. Excepto el comando de ayuda, todos los comandos se escriben en mayúsculas.

Los comandos de iptables son los siguientes:

- -A — Añade la regla iptables al final de la cadena especificada. Éste es el comando utilizado para simplemente añadir una regla cuando el orden de las reglas en la cadena no importa.
- -C — Verifica una regla en particular antes de añadirla en la cadena especificada por el usuario. Este comando puede ser de ayuda para construir reglas iptables complejas pidiéndole que introduzca parámetros y opciones adicionales.
- -D — Borra una regla de una cadena en particular por número (como el 5 para la quinta regla de una cadena). Puede también teclear la regla entera e iptables borrará la regla en la cadena que corresponda.
- -E — Renombra una cadena definida por el usuario. Esto no afecta a la estructura de la tabla. Tan solo le evita el problema de borrar la cadena, creándola bajo un nuevo nombre, y reconfigurando todas las reglas de dicha cadena.
- -F — Libera la cadena seleccionada, que borra cada regla de la cadena. Si no se especifica ninguna cadena, este comando libera cada regla de cada cadena,
- -h — Proporciona una lista de estructuras de comandos útiles, así como una resumen rápido de parámetros de comandos y opciones.

- -I — Inserta una regla en una cadena en un punto determinado. Asigne un número a la regla a insertar e iptables lo pondrá allí. Si no especifica ningún número, iptables posicionará su comando al principio de la lista de reglas.

> **Nota importante**
>
> Tenga cuidado con qué opción (-A o -I) está usando al añadir una regla. El orden de las reglas puede ser muy importante cuando esté determinando si se aplica una regla u otra a un paquete en particular. Asegúrese al añadir una regla al principio o al final de la cadena de que no afecta a otras reglas de la misma cadena.

- -L — Lista todas las reglas de la cadena especificada tras el comando. Para ver una lista de todas las cadenas en la tabla filter por defecto. La sintaxis siguiente deberá utilizarse para ver la lista de todas las reglas de una cadena específica en una tabla en particular:

  iptables -L <*Cain-name*> -t <*table-name*>

- -N — Crea una nueva cadena con un nombre especificado por el usuario.
- -P — Configura la política por defecto para una cadena en particular de tal forma que cuando los paquetes atraviesen la cadena completa sin cumplir ninguna regla, serán enviados a un objetivo en particular, como puedan ser ACCEPT o DROP.
- -R — Reemplaza una regla en una cadena en particular. Deberá utilizar un número de regla detrás del nombre de la cadena para reemplazar esta cadena. La primera regla de una cadena se refiere a la regla número 1.
- -X — Borra una cadena especificada por el usuario. No se permite borrar ninguna de las cadenas predefinidas para cualquier tabla.
- -Z — Pone ceros en los contadores de byte y de paquete en todas las cadenas de una tabla en particular.

Parámetros

Una vez que se hayan especificado algunos comandos de iptables, incluyendo aquellos para crear, añadir, borrar, insertar o reemplazar reglas de una cadena en particular, se necesitan parámetros para comenzar la construcción de la regla de filtrado de paquetes.

- -c Resetea los contadores de una regla en particular. Este parámetro acepta las opciones PKTS y BYTES para especificar qué contador hay que resetear.
- -d Configura el nombre de la máquina destino, dirección IP o red de un paquete que cumplirá la regla. Cuando se especifique una red, puede utilizar dos métodos diferentes para describir la máscara de red, como 172.16.0.28/255.255.255.0 o 172.16.0.28/24.
- -f Aplica esta regla solo a los paquetes fragmentados.
- Usando la opción ! después de este parámetros, únicamente los paquetes no fragmentados se tendrán en cuenta.
- -i Configura las interfaces de entrada de red, como eth0 o ppp0, para ser usadas por una regla en partículas. Con iptables, este parámetro opcional sólo debería de ser usado por las cadenas INPUT y FORWARD cuando se utilice junto con la tabla filter y la cadena PREROUTING con las tablas nat y mangle.
  Éste parámetro proporciona varias opciones útiles que pueden ser usadas antes de especificar el nombre de una interfaz:
  - ! — Dice a este parámetro que no concuerde, queriendo decir esto que las interfaces especificadas se excluirán de esta regla.
  - + — Caracter comodín usado para hacer coincidir todas las interfaces que concuerden con una cadena en particular. Por ejemplo, el parámetro -i eth+ aplicará esta regla a todas las interfaces Ethernet de su sistema excluyendo cualquier otro tipo de interfaces, como pueda ser la ppp0.

Si el parámetro -i se utiliza sin especificar ninguna interfaz, todas las interfaces estarán afectadas por la regla.

- -j Dice a iptables que salte a un objetivo en particular cuando un paquete cumple una regla en particular. Los objetivos válidos que se usarán tras la opción -j incluyen opciones estándar, ACCEPT, DROP, QUEUE, y RETURN, así como opciones extendidas que están disponibles a través de módulos que se cargan por defectos con el paquete RPM de iptables de Red Hat Linux, como LOG, MARK, y REJECT, así como otras. Mire la página del manual de iptables para obtener más información sobre este y otros muchos objetivos, incluyendo reglas de ejemplo que los utilizan, así como objetivos que pueden ser usados solamente en una tabla en particular.

En lugar de especificar la acción objetivo, puede también dirigir un paquete que cumpla la regla hacia una cadena definida por el usuario fuera de la cadena actual. Esto le permitirá aplicar otras reglas contra este paquete, y filtrarlo mejor con respecto a otros criterios.

Si no especifica ningún objetivo, el paquete se mueve hacia atrás en la regla sin llevar a cabo ninguna acción. A pesar de todo, el contador para esta regla se sigue incrementando en uno, a partir del momento en el que el paquete se adecua a la regla especificada.

- -o Configura la interfaz de red de salida para una regla en particular, y sólo puede ser usada con las cadenas OUTPUT y FORWARD en la tabla filter y la cadena POSTROUTING en las tablas nat y mangle. Estas opciones de los parámetros son los mismos que para los de la interfaz de red de entrada (opción -i).
- -p Configura el protocolo IP para la regla, que puede ser icmp, tcp, udp, o all(todos), para usar cualquier protocolo. Además, se pueden usar otros protocolos menos usados de los que aparecen en /etc/protocols. Si esta opción se omite al crear una regla, la opción all es la que se selecciona por defecto.
- -s Configura el origen de un paquete en particular usando la misma sintaxis que en el parámetro de destino (opción -d).

Opciones de identificación de paquetes

Los diferentes protocolos de red proporcionan opciones especializadas de concordancia que pueden ser configurados de forma específica para identificar un paquete en particular usando dicho protocolo. Por supuesto el protocolo deberá ser especificado en un primer momento en el comando iptables, como con la opción -p tcp *<noñbre-protocolo>*, para hacer que las opciones de dicho protocolo estén disponibles.

Protocolo TCP

Estas opciones de identificación están disponibles en el protocolo TCP (opción -p tcp):

- --dport Configura el puerto de destino para el paquete. Puede utilizar un nombre de servicio de red (como www o smtp), un número de puerto, o bien un rango de números de puertos para configurar esta opción. Para ver los nombres o alias de los servicios de red y los números de puertos que usan mire el fichero /etc/services. Puede usar la opción --destination-port para especificar esta opción de identificación de paquete. Para especificar un rango de números de puertos separe los dos números de puertos con dos puntos (:), como en -p tcp --dport 3000:3200. El rango válido más grande es 0:65535. También puede usar el carácter de punto de exclamación (!) como flag tras la opción --dport para decirle a iptables que seleccione los paquetes que *no* usen ese servicio o puerto.
- --sport Configura el puerto de origen del paquete, usando las mismas opciones que --dport. También puede usar --source-port para especificar esta opción.
- --syn Provoca que todos los paquetes designados de TCP, comúnmente llamados *paquetes SYN*, cumplan esta regla. Cualquier paquete que esté llevando un payload de datos no será tocado. Si se sitúa un punto de exclamación (!) como flag tras la opción --syn se provoca que todos los paquetes no-SYN sean seleccionados.
- --tcp-flags Permite que los paquetes TCP con conjuntos de bits específicos, o flags, sean seleccionados para una regla. La opción de selección --tcp-flags

acepta dos parámetros, que son los flags para los diferentes bits ordenados en una lista separada por comas. El primer parámetro es la máscara, que configura los flags que serán examinados en el paquete. El segundo parámetro se refiere a los flags que se deben configurar en el paquete para ser seleccionado. Los flags posibles son ACK, FIN, PSH, RST, SYN y URG. Adicionalmente, se pueden usar ALL y NONE para seleccionar todos los flags o ninguno de ellos.

- Por ejemplo, una regla iptables que contiene -p tcp --tcp-flags ACK,FIN,SYN SYN tan solo seleccionará los paquetes TCP que tengan el flag SYN activo y los flags ACK y FIN sin activar.
- Como en otras opciones, al usar el punto de exclamación (!) tras --tcp-flags invierte el efecto de la opción, de tal forma que los flags del parámetro no tendrán que estar presentes para poder ser seleccionados.
- --tcp-option Intenta seleccionar con opciones específicas de TCP que pueden estar activas en un paquete en particular. Esta opción se puede revertir con el punto de exclamación (!).

Protocolo UDP

Estas opciones de selección están disponibles para el protocolo UDP (-p udp):

- --dport Especifica el puerto destino del paquete UDP usando el nombre del servicio, el número del puerto o un rango de puertos. La opción de selección de paquetes --destination-port se puede utilizar en lugar de --dport.
- --sport Especifica el puerto origen del paquete UDPm usando el nombre del servicio número de puerto o rango de puertos. La opción --source-port puede ser usada en lugar de --sport.

Protocolo ICMP

Los paquetes que usan el protocolo de control de mensajes de Internet (Internet Control Message Protocol, ICMP) pueden ser seleccionados usando la siguiente opción cuando se especifique -p icmp:

- --icmp-type Selecciona el nombre o el número del tipo ICMP que concuerde con la regla. Se puede obtener una lista de nombres válidos ICMP tecleando el comando iptables -p icmp -h.

Módulos con opciones de selección adicionales

Las opciones de selección adicionales, que no son específicas de ningún protocolo en particular están también disponibles a través de módulos que se cargan cuando el comando iptables los necesite. Para usar una de estas opciones deberá cargar el módulo por su nombre incluyendo -m *<nombre-modulo>* en el comando iptables que crea la regla.

Un gran número de módulos, cada uno de ellos con sus diferentes opciones de selección de paquetes están disponibles por defecto. También es posible crear sus propios módulos que proporcionen funcionalidades de selección adicionales, puede que para requisitos específicos de su red. Existen muchos módulos, pero tan solo los más populares serán vistos aquí.

El módulo limit le permite poner un límite en el número de paquetes que podrán ser seleccionados por una regla en particular. Esto es especialmente beneficioso cuando se usa la regla de logging ya que hace que el flujo de paquetes seleccionados no llene nuestros ficheros log con mensajes repetitivos ni utilice demasiados recursos del sistema.

- --limit — Configura el número de coincidencias en un intervalo de tiempo, especificado con un número y un modificador de tiempo ordenados en el formato *<número>/<tiempo>*. Por ejemplo, si usamos --limit 5/hour solo dejaremos que una regla sea efectiva cinco veces a la hora,

- Si no se utiliza ningún número ni modificador de tiempo, se asume el siguiente valor por defecto: 3/hour.
- --limit-burst — Configura un límite en el número de paquetes capaces de cumplir una regla en un determinado tiempo. Esta opción deberá ser usada junto con la opción --limit, y acepta un número para configurar el intervalo de tiempo (threshold).
- Si no se especifica ningún número, tan solo cinco paquetes serán capaces inicialmente de cumplir la regla.
- El módulo state, utiliza la opción --state, puede seleccionar un paquete con los siguientes estados de conexión particulares:
- ESTABLISHED El paquete seleccionado se asocia con otros paquetes en una conexión establecida.
- INVALID El paquete seleccionado no puede ser asociado a una conexión conocida.
- NEW El paquete seleccionado o bien está creando una nueva conexión o bien forma parte de una conexión de dos caminos que antes no había sido vista.
- RELATED El paquete seleccionado está iniciando una nueva conexión en algún punto de la conexión existente.
- Estos estados de conexión se pueden utilizar en combinación con otros separándolos mediante comas como en -m state --state INVALID,NEW.
- Para seleccionar una dirección MAC hardware de un dispositivo Ethernet en particular utilice el módulo mac, que acepta --mac-source con una dirección MAC como opción. Para excluir una dirección MAC de una regla, ponga un punto de exclamación (!) tras la opción --mac-source.
- Para ver otras opciones disponibles a través de módulos, mire la página del manual de iptables.

Opciones del objetivo

Una vez que un paquete cumple una regla en particular, la regla puede dirigir el paquete a un número de objetivos (destinos) diferentes que decidirán cuál será su destino y, posiblemente, las acciones adicionales que se tomarán, como el guardar un registro de lo que está ocurriendo. Adicionalmente, cada cadena tiene un objetivo por defecto que será el que se utilice si ninguna de las reglas disponibles en dicha cadena se pueden aplicar a dicho paquete, o si ninguna de las reglas que se aplican al mismo especifican un objetivo concreto.

Existen pocos objetivos standard disponibles para decidir qué ocurrirá con el paquete:

- *<user-defined-chain>* — El nombre de una cadena que ya ha sido creada y definida con anterioridad junto con esta tabla con reglas que serán verificadas contra este paquete, además de cualquier otra regla en otras cadenas que se deban verificar contra este paquete. Este tipo de objetivo resulta útil para escrutinar un paquete antes de decidir qué ocurrirá con él o guardar información sobre el paquete.
- ACCEPT — Permite que el paquete se mueva hacia su destino (o hacia otra cadena, si no ha sido configurado ningún destino ha sido configurado para seguir a esta cadena).
- DROP — Deja caer el paquete al suelo. El sistema que envió el paquete no es informado del fallo. El paquete simplemente se borra de la regla que está verificando la cadena y se descarta.
- QUEUE — El paquete se pone en una cola para ser manejado por una aplicación en el espacio de usuario.
- RETURN — Para la verificación del paquete contra las reglas de la cadena actual. Si el paquete con un destino RETURN cumple una regla de una cadena llamada desde otra cadena, el paquete es devuelto a la primera cadena para retomar la verificación de la regla allí donde se dejó. Si la regla RETURN se utiliza en una cadena predefinida, y el paquete no puede moverse hacia la cadena anterior, el objetivo por defecto de la cadena actual decide qué acción llevar a cabo.

Además de estos objetivos standard, se pueden usar otros más con extensiones llamadas *módulos de objetivos* (target modules), que trabajan de forma similar a como los hacían los módulos de las opciones de selección..

Existen varios módulos extendidos de objetivos, la mayoría de los cuales sólo se aplicarán a tablas o situaciones específicas. Un par de estos módulos de los más populares e incluidos por defecto en Red Hat Linux serían:

- LOG Guarda un registro de todos los paquetes que cumplen esta regla. Como estos paquetes son monitorizados por el kernel, el fichero /etc/syslog.conf determina dónde se escribirán esas entradas en el fichero de registro (log). Por defecto, se sitúan en el fichero /var/log/messages.
  Se pueden usar varias opciones tras el objetivo LOG para especificar la manera en la que tendrá lugar el registro:
  - --log-level Configura un nivel de prioridad al evento de registro del sistema. Se puede encontrar una lista de los eventos del sistema en la página del manual de syslog.conf, y sus nombres se pueden usar como opciones tras la opción --log-level.
  - --log-ip-options Cualquier opción en la cabecera de un paquete IP se guarda en el registro.
  - --log-prefix Pone una cadena de texto antes de la línea de registro cuando ésta sea escrita. Acepta hasta 29 caracteres tras la opción --log-prefix. Esto puede ser útil para escribir filtros del registro del sistema para ser usados conjuntamente junto con el registro de paquetes.
  - --log-tcp-options Cualquier opción en la cabecera de un paquete TCP se guarda en el registro.
  - --log-tcp-sequence Escribe le número de secuencia TCP del paquete en el registro del sistema.

REJECT Envía un paquete de error de vuelta al sistema que envió el paquete, y lo deja caer (DROP). Este objetivo puede ser útil si queremos notificar al sistema que envió el paquete del problema.

El objetivo REJECT acepta una opción --reject-with <*type*> que permite que información más detallada sea enviada junto con el paquete de error. El mensaje port-unreachable es el error <*type*> que se envía por defecto cuando no se utiliza junto con otra opción. Para obtener una lista completa de todas las opciones <*type*> que se pueden utilizar, vea la página del manual de iptables.
Podrá encontrar otras extensiones de objetivos incluyendo algunas muy útiles con masquerading usando la tabla nat o con alteración de paquetes usando la tabla mangle, en la página del manual de iptables.

Opciones de listado

El comando de listado por defecto, iptables -L, proporciona una visión básica de las cadenas actuales de la tabla de filtros por defecto. . Existen opciones adicionales que proporcionan más información y la ordenan de diferentes formas:

- -v Muestra la salida por pantalla, como el número de paquetes y bytes que cada cadena ha visto, el número de paquetes y bytes que cada regla ha encontrado, y qué interfaces se aplican a una regla en particular.
- -x Expande los números en sus valores exactos. En un sistema ocupado, el número de paquetes y bytes vistos por una cadena en concreto o por una regla puede estar abreviado usando K (miles), M (millones), y G (billones) detrás del número. Esta opción fuerza a que se muestre el número completo.
- -n Muestra las direcciones IP y los números de puertos en formato numérico, en lugar de utilizar el nombre del servidor y la red tal y como se hace por defecto.
- --line-numbers Proporciona una lista de cada cadena junto con su orden numérico en la cadena. Esta opción puede ser útil cuando esté intentando borrar una regla específica en una cadena, o localizar dónde insertar una regla en una cadena.

Guardar información de iptables

Las reglas creadas con el comando iptables se almacenan solamente en RAM. Si tiene que reiniciar su sistema tras haber configurado reglas de iptables, éstas se perderán. Si quiere que determinadas reglas de filtro de red tengan efecto en cualquier momento que inicie su sistema, necesita guardarlas en el fichero /etc/sysconfig/iptables. Para hacer esto teclee el comando /sbin/service iptables save como usuario root. Esto hace que el script de inicio de iptables ejecute el programa /sbin/iptables-save y escriba la configuración actual de iptables en el fichero /etc/sysconfig/iptables. Este fichero debería ser de solo lectura para el usuario root, para que las reglas de filtrado de paquetes no sean visibles por el resto de los usuarios.

La próxima vez que se inicie el sistema, el script de inicio de iptables volverá a aplicar las reglas guardadas en /etc/sysconfig/iptables usando el comando /sbin/iptables-restore.

Mientras que siempre es una buena idea el verificar cada nueva regla de iptables antes de que se escriba en el fichero /etc/sysconfig/iptables, es posible copiar las reglas de iptables en este fichero a partir de otra versión del sistema de este fichero. Esto le permitirá distribuir rápidamente conjuntos de reglas iptables a diferentes máquinas.

Nota importante

Si distribuye el fichero /etc/sysconfig/iptables a otras máquinas, debe escribir /sbin/service iptables restart para que las nuevas reglas tengan efecto.

Lo que hasta aquí se ha mencionado sólo es un resumen de lo que puede encontrar en el manual de iptables que contiene una descripción detallada de los diversos comandos, parámetros y otras opciones que podrán ayudarle para añadir nuevas tablas y construir reglas para las cadenas.

Sitios web útiles

- http://netfilter.samba.org/ — Contiene información diversa sobre iptables, incluyendo las FAQ sobre problemas específicos que le pueden ocurrir así como varias guías de ayuda escritas por Rusty Russell, el mantenedor del cortafuegos IP de Linux. La documentación y los COMO cubren temas como conceptos básicos de red, el filtrado de paquetes en los kernels 2.4, configuración de NAT.
- http://www.linuxnewbie.org/nhf/Security/IPtables_Basics.html — una visión básica y general sobre la forma en la que los paquetes se mueven dentro del kernel de Linux, además de una introducción sobre cómo se construyen comandos iptables simples.
- http://www.redhat.com/support/resources/networking/firewall.html — Esta página contiene enlaces actualizados a una gran variedad de recursos para el filtrado de paquetes.

Firewalls

Los Firewalls para filtrar paquetes y proxies que bloquean intromisiones no deseadas en el sistema.

Ipchains
Redir
Tinyproxy

Ipchains

La utilidad ipchains es un software de filtrado de paquetes estándar que está disponible en Red Hat Linux. Permite establecer las reglas según las cuales serán filtrados los paquetes antes de que se autorice su entrada o salida del sistema.

Redir

Redireccionamiento de puertos redir. Su función es entender a un puerto en particular, cuando recibe una conexión, la encamina a otro puerto y transmite los datos entre ambos. La misma función se encuentra en la utilidad ssh. Redir se localiza en http://wwwusers.qual.net/(sammy/hacks/.

Tinyproxy

Es un proxy http pequeño pero muy efectivo, que resulta útil cuando squid resulta demasiado pesado o un riesgo para la seguridad. Se puede obtener de http://flarenet.com/tinyproxy/.

Lista de algunos Firewalls

Netfilter
Check Point Firewall-1
FWTK
PIX
SOCKS
Squid
PBX firewall
IPCHAINS
RaptorFirewall

APÉNDICE B

Herramientas de seguridad y monitoreo, servicios de red UNIX.

XINETD

Control del acceso con el comando xinetd

Los beneficios de que ofrecen los wrappers TCP son todavía mayores cuando se usa la librería libwrap.a junto con el comando xinetd, un *súper-demonio* que ofrece acceso adicional, conexión, vinculación, redirección y control del uso de los recursos.

Ficheros de configuración del comando xinetd

El servicio xinetd lo controla el fichero / etc/xinetd.conf así como otros ficheros específicos que se encuentran en el directorio /etc/xinetd.d/

/etc/xinetd.conf

El fichero xinetd.conf es el padre de todos los ficheros de configuración del comando xinetd ya que todos los ficheros de servicios determinados también se sintetizan cada vez que se activa el comando xinetd. Por defecto, el fichero xinetd.conf contiene pautas de configuración básicas que se aplican a cada servicio.

Control del acceso con el comando xinetd

Los usuarios de los servicios xinetd pueden elegir los ficheros de control de acceso con los wrappers TCP (/etc/hosts.allow y /etc/hosts.deny) que ofrecen el control del acceso vía los ficheros de configuración xinetd o la mezcla de ambos.

El comando xinetd para el control del acceso disponible en los diversos ficheros de configuración es distinto para cada uno de los métodos que usan los wrappers TCP. Mientras que los wrappers TCP sitúan toda la configuración del acceso en dos archivos /etc/hosts.allow y /etc/hosts.deny, cada uno de los ficheros del servicio que se

encuentran en el directorio /etc/xinetd.d pueden contener las reglas de control del acceso que se basan en las máquinas que están autorizadas para usar dicho servicio.

Las páginas man xinetd y xinetd.conf contienen información adicional para la creación de ficheros de configuración xinetd y una descripción del funcionamiento del comando xinetd.

Sitios Web útiles

http://www.xinetd.org/, xinetd, contiene ejemplos de ficheros de configuración, una lista completa de características y FAQs informativas.

http://www.macsecurity.org/resources/xinetd/tutorial.shtml, trata las distintas maneras de usar los ficheros de configuración xinetd para alcanzar determinados objetivos de seguridad.

TCP WRAPERS

Los wrappers TCP

El control del acceso a los servicios de red puede ser todo un reto. Los firewalls controlan el acceso a una determinada red pero son difíciles de configurar. Para facilitar esta tarea existen los wrappers TCP y el comando xinetd controlan el acceso a los servicios basándose en las direcciones IP y en los nombres de las máquinas. Además, estas herramientas también incluyen las funciones de administración y de funcionamiento que son fáciles de configurar.

¿Qué son los *wrappers* TCP?

Wrappers TCP están instalados por defecto en la instalación de tipo servidor del sistema operativo Red Hat Linux *8.0* y proporcionan control de acceso a gran variedad de servicios. Muchos servicios modernos de red como por ejemplo SSH, Telnet y FTP usan los *wrappers TCP*, un programa que ha sido diseñado para controlar las peticiones entrantes y los servicios requeridos.

Los wrappers TCP están basados en la idea de que más que permitir la conexión entrante de un cliente directamente con un demonio de servicios de red que se ejecuta como un proceso separado en un sistema de servidores. Éstos "capturan" la petición protegiéndola y permitiendo así un mayor control del acceso y de conexión al usuario que esté intentando usar el servicio.

La librería que se encarga de esta funcionalidad es libwrap.a, una librería que administra servicios de red como xinetd, sshd y portmap que ya están compilados. También puede compilar otros servicios de red e incluso escribir programas de red con la librería libwrap.a. Red Hat Linux proporciona los programas de los *wrappers* TCP necesarios en el fichero RPM *tcp_wrappers-<version>*.

Ventajas de los wrappers TCP

Cuando alguien intenta acceder a un servicios de red con los *wrappers* TCP, un programa pequeño de protección da el nombre del servicio requerido y la información de la máquina cliente. Este programa no devuelve ninguna información al cliente y una vez que se cumplen las directivas del control de acceso el programa desaparece evitando así una sobrecarga en la comunicación entre el cliente y el servidor.

Los *wrappers* TCP ofrecen dos ventajas básicas comparado con las otras técnicas de control de servicios de red:

- *El cliente que se conecta no sabe que se están usando los wrappers TCP.* Los usuarios legales no notarán ninguna diferencia y los invasores nunca reciben información adicional sobre el motivo por el que las conexiones realizadas fallaron.
- *Los wrappers TCP funcionan independientemente de las aplicaciones que el programa.* Esto permite que muchas aplicaciones puedan compartir un set de ficheros de configuración común facilitando la administración.

Herramientas para detectar intrusiones

Existen muchas herramientas para detectar intrusiones que pueden ser utilizadas para asegurar que no se pasan por alto las señales de intrusos en la red. Todas estas herramientas, aunque son muy útiles, no pueden reemplazar la comprobación de registros hecha periódicamente por uno mismo.

Analizador(es) del sistema de archivos

Herramienta de utilidad para la comprobación de intrusiones son los analizadores del sistema de archivos. Entre ellas se encuentran paquetes como los siguientes:

Tripwire.
Se trata de un producto que analiza el sistema de archivos y registra firmas digitales para los archivos que encuentra. Se puede ejecutar periódicamente para comparar los archivos con las firmas registradas y los cambios en los indicadores. Otra forma de conseguir resultados idénticos es almacenar todos los archivos de configuración del sistema en un deposito y ejecutar periódicamente la utilidad dic de CVS tomando nota

de los cambios encontrados. El inconveniente es que CVS está mejor dotado para los archivos de texto, por lo que no es una elección adecuada para encontrar diferencias en los ejecutables del sistema. Otro tipo de cambio fácilmente detectable es la existencia de cambios en los permisos del archivo, especialmente los bit setuid o setgid.
No deben descartarse las herramientas estándar como who, top, ls y ps. Aunque estas herramientas son a menudo modificadas por los crakers, es posible ir un paso delante de ellos manteniendo copias de estas utilidades.

Trinux

Una herramienta muy útil que se puede utilizar en caso de una brecha en el sistema es la distribución de Linux llamada Trinux. Se trata de una distribución de Linux soportada en discos que usa RAM para proporcionar una amplia colección de herramientas de seguridad y recuperación, en un espacio mínimo.

Una de las razones más poderosas para utilizar Trinux es que se puede arrancar el sistema afectado usando los disquetes de Trinux y usar las herramientas que se incluyen para diagnosticar el sistema. Esto permite investigar sin afectar al estado del sistema y evitando las herramientas del sistema, que pueden haber sido alteradas por el intruso. Se puede conseguir una copia de Trinux, visitando el sitio *web* http://www.trinux.org y descargando las últimas imágenes y paquetes.

Rastreador de paquetes

Son herramientas que están a la escucha de una interfaz y reunen información sobre paquetes que pasan por la misma. Las tres utilidades de este tipo incluidas en Trinux son:

Tcpdump
Lpgrab
Ngrep

Tcpdump

Es el rastreador de paquetes estándar que viene incluido en la mayoría de las versiones UNIX y Linux. La información del manual asegura que Tcpdump imprime las cabeceras de los paquetes que coinciden con la expresión booleana suministrada en la línea de comando. Este paquete se encuentra disponible por FTP anónimo en ftp://ftp.ee.lbl.gov/tcpdump.tar.Z

Ipgrab

Ipgrab es otro rastreador .de paquetes basado en la biblioteca de captura de paquetes de Berkeley. Imprime información completa sobre la cabecera para paquetes en los niveles de enlace, red y transporte. Está escrito por Mike Botella y se encuentra en la *Web* en http://www.xnet.com/-cathmike/MSB/Ssoftware/.

Ngrep

La última herramienta de esta categoría es ngrep. Se trata de un intento de proporcionar funciones semejantes a las del Grez de GNU a través del nivel de red. Permite especificar una expresión regular y filtrar las expresiones tal y como son entendidas por la biblioteca de captura de paquetes de Berkeley, imprimiendo información sobre los paquetes coincidentes. Se encuentra en http://www.packetfactory.net/ngrep.

Monitores de red

Los monitores de red se usan para exponer información estadística acerca del tráfico que transita por una red. Este tipo de herramientas se puede usar para determinar la cantidad y tipo de trafico, incluyendo recuentos de bytes, número y tipo de paquetes, recuento de errores y otra información similar. Estos son dos de las herramientas distribuidas con Trinux:

IPTraf

ntop

IPTraf

Se trata de un paquete de estadísticas de red que muestra información general y estadísticas detalladas mostrando recuento de paquetes, tamaños y muchos otros datos referentes al tráfico en la red en modo de pantalla completa. Incluye un mecanismo de filtrado que permite seleccionar por protocolo tipos de tráfico específicos. El paquete IPTraf está disponible en http://cebu.mozcom.com/riker/iptraf/.

ntop

ntop es otro paquete de vigilancia que muestra la información de manera semejante a la utilidad top y, como ésta, muestra estadísticas de utilización de la red ordenadas de diversas formas. Se puede utilizar en modo interactivo o en modo *web*. Se encuentra disponible en http://www.serra.inipi.it/-ntop/.

Mapeo de red/análisis de puntos débiles

Las herramientas de mapeo de red y análisis de puntos débiles usan una gran variedad de métodos para determinar el trazado de una red. Comprueban varios servicios bien conocidos y usan las respuestas para identificar los puertos abiertos. O bien, mediante el uso de técnicas tales como TCP fingerprinting (impresión de huellas TCP), determinan el tipo de sistema al que llegan. Trinux incluye varias herramientas de este tipo, como por ejemplo:

nmap.
Exscan.
SAINT.
QueSo.
Hping.
Firewalk.
Cgichk.

nmap

Es una de las herramientas más útiles con la que cuenta Trinux. Explora las redes e identifica los tipos de hosts atacados y los posibles puntos vulnerables de cada uno de ellos. Nmap es una herramienta de auditoria excelente y debería formar parte del cuadro de herramientas de cualquier administrador que se aprecie. Está disponible en http://www.insecure.org/nmap/index.html.

Exscan

Es otra utilidad de análisis. Lo mismo que nmap, puede identificar el sistema operativo del sistema que está siendo analizado a través del uso de TCP fingerprinting, una característica que obtiene mediante la integración con queso. Además, captura la información de varios servicios en sistemas remotos como SMTP y FTP y busca en los servidores información como la versión http o información de finger. Se puede conseguir en:
http://exscan.netpedia.net/exscan.html.

SAINT.

SAINT (Security Administrator's Integrated Network Tool, herramienta de red integrada para el administrador de seguridad) analiza los sistemas buscando puntos inseguros y puertos sin protección. También puede examinar los puntos vulnerables relacionados con una relación de confianza entre sistemas. Se puede encontrar en http://www.wwwdsi.com/.

Hping

Hping es una utilidad destinada a la auditoria de las pilas de TCP/IP, descubriendo la política de cortafuegos, y analizando los puertos TCP de varias maneras, entre otras cosas. Al contrario que las utilidades *ping* normales, *ping* realiza los *pings* TCP en lugar de ICMP. Se puede descargar desde http://www.kyuzz.org/antirez/hping2.html

Firewalk

La utilidad Firewalk es una herramienta de análisis similar a traceroute y utiliza las respuestas de los sistemas para determinar las listas de control de acceso a pasarelas. Puede usarse para determinar las reglas de filtrado efectivas en un dispositivo de reenvío de paquetes como un encaminador. Se puede encontrar en http://packetfactory.net/firewalk

Cgichk.

Se trata de un analizador sencillo disponible en http://www.rootshell.com/, que conecta a un servidor *web* y busca programas CGI reconocidamente vulnerables. Este programa busca aquellos que han sido identificados por sufrir fallos de seguridad.

Servicios de UNIX

Esta tabla (Tabla A1) describe algunos servicios de red disponibles en los diferentes niveles de ejecución.

Demonio	Descripción	Consideraciones y recomendaciones
amd	Habilita el demonio de montaje automático de NFS, amd.	NFS se diseño como un servicio LAN Contiene numerosas debilidades de seguridad si se usa sobre Internet. No permite acceso a Internet a los. archivos montados NFS No ejecute amd en la máquina Firewall
autofs	Habilita el proceso de administración de montaje automático, *automount*.	NFS es un servicio LAN basado en RPC. Tanto NFS como cualquier servicio de red se basa en el demonio *portmap* son posibles agujeros de seguridad. No se ejecute automount en la máquina Firewall.
bootparamd	Habilita el servidor de parámetros de inicio.	El demonio bootparamd proporciona información relacionada con el inicio de estaciones sin disco sobre una LAN. No ejecute bootparamd.
dhcpd	Inicio un servidor DHCPD local.	Este servicio asigna direcciones IP asignadas dinámicamente a los host clientes. No ejecute DHCP a menos que realmente lo necesite.
gated	Habilita el demonio de enrutamiento de pasarela.	El demonio gated maneja los protocolos de enrutamiento de red. No ejecute gated.
httpd	Inicia el servidor web	Apache para albergar un sitio web. Ejecútelo si lo necesita.
inet	El demonio inetd es el fundamento para proporcionar muchos servicios de red. En lugar de tener, como mínimo, un demonio de cada servicio ejecutándose contínuamente.	Es necesario inetd si se usa servicios comunes como ftp o telnet, localmente, o si se ofrece estos servicios a sitios remotos. Ejecute inetd.

Tabla A1. Servicios de Unix

Demonio	Descripción	Consideraciones
linuxconf	Permite configurar la máquina usando un servidor *web* local como interfaz de usuario.	linuxconf escucha el puerto 98 TCP. De forma predeterminada solo escucha la interfaz de bucle invertido.
lpd	Habilita el servidor de impresión.	Con UNIX, las impresoras retratan como dispositivos de red. Asegurese de que los archivos de configuración de acceso a la impresora bloqueen el acceso remoto.
named	El demonio named proporciona la mitad del servidor DNS de red, traduciendo entre nombres de máquina simbólicos y sus direcciones IP numéricas	La parte del cliente DNS, la que resuelve las traducciones, no es visible como programa independiente. Es parte de las bibliotecas de red compiladas dentro de los programas.
netfs	Monta sistemas de archivos en red NFS, Samba y NetWare.	netfs no es un demonio de servicio. Es un guión de *shell* que se ejecuta una vez para montar los sistemas de archivos conectados en red de forma local.
network	La secuencia de comandos de configuración network se ejecuta en tiempo de inicio para activar las interfaces de red que se han configurado.	Se debe de ejecutar esta secuencia de comandos. Ejecute network.
nfs	Habilita servicios NFS.	NFS se diseño como un servicio LAN. Contiene una gran cantidad de debilidades se usa sobre Internet. NFS lo forman varios demonios, los cuales son *amd, nfs y nfsfs.* No ejecute nfs en la máquina Firewall.
nscd	Habilita el demonio Name Switch Cache.	*nscd* es un servicio para compatibilidad NIS que introduce en la caché las contraseñas de usuario y los miembros del grupo. No ejecute nscd en la máquina Firewall.

Tabla A1. Servicios de Unix (continuación)

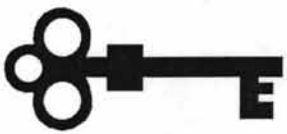
Demonio	Descripción	Consideraciones
routed	Habilita el demonio routed para actualizar automáticamente el núcleo dinámico de las tablas de enrutamiento.	routed representa serios problemas de seguridad. Es recomendable usar simplemente direccionamiento IP estático localmente. No ejecute routed.
rstatd	Habilita el demonio rstatd para coleccionar y proporcionar información del sistema para otras máquinas de la LAN.	La información de estado del sistema no debe compartirse con las máquinas de Internet remotas. No ejecute rstatd en la máquina Firewall.
ruserd	Habilita el servicio de localización de usuarios. Este es un servicio basado en RPC que ofrece información de usuarios individuales que tienen actualmente una sesión abierta de una las máquinas de la LAN.	Un sitio pequeño no tiene necesidad de este servicio LAN. No ejecute ruserd en la máquina Firewall.
rwhod	Habilita el demonio de servicio rwhod. El demonio rwhod es compatible con los servicios *rwho* y *ruptime* para una LAN. Como tales, el servicio ofrece información sobre quién tiene una sesión iniciada, que están haciendo, qué sistemas se están ejecutando y están conectados a la LAN, etc.	Un sitio pequeño no tiene necesidad de este servicio LAN. No ejecute rwhod en la máquina Firewall.
sendmail	El servicio de correo local se controla mediante sendmail	sendmail es necesario si alberga servicios propios de correo. Correctamente configurado, actualmente sendmail es relativamente seguro.
smb	Permite el servicio Samba para compartir archivos, así como para compartir impresoras.	No se levante este servicio si su LAN no cuenta con máquinas Windows. No ejecute smb en la máquina Firewall.

Tabla A1. Servicios de Unix (continuación)

Demonio	Descripción	Consideraciones
snmpd	Habilita el demonio simple de administración de red. El demonio snmpd controla la administración de red SNMP.	SNMP es un servicio de administración LAN. Si es necesario usarlo debe considerarse como un servicio peligroso y se debe bloquear todo el trafico entre la LAN e Internet. No se ejecute snmpd.
squid	Habilita la *Squid internet Object Cache*. Si no se ejecuta el servidor *web* Apache localmente, squid puede servir como un servidor proxi HTTP local y como caché *web* local para las páginas *web* obtenidas de sitios remotos.	Si se configura correctamente, squid no implica especiales consideraciones de seguridad. Ejecute squid si lo necesita.
syslog	La secuencia de comandos de configuración syslog inicia los demonios de registro del sistema *syslogd* y *klogd* en tiempo de inicio.	Este servicio es necesario para que el estado del sistema y los mensajes de error se escriban en los archivos de registro, *syslogd* puede configurarse para ser ejecutado como un servicio LAN. Ejecute syslog.
xfs	Habilita el servidor de fuentes de X Windows.	En su configuración predeterminada xfs escucha en un socket de dominio UNIX privado. Como tal, xfs no representa en si un riesgo para la seguridad. El servidor X Window depende de xfs. Idealmente, el servicio no debe ejecutarse en una máquina Firewall. Si es posible no ejecutarlo en la máquina Firewall.
xntpd	Habilita un servidor de tiempo de red local.	El servidor **xntpd** local se ejecuta para distribuir la hora del sistema actual entre máquinas locales de una LAN interna.

Tabla A1. Servicios de Unix (continuación)

APÉNDICE C



Políticas y reglas del Firewall

POLÍTICAS DE SEGURIDAD

Políticas de conexión al sistema o de control de acceso

- El administrador deberá conectarse a los servidores a través de programas seguros (*ssh, scp o sftp*).
- El administrador deberá accesar a los servidores de producción a través de una dirección IP definida, preferentemente pertenecientes a las asignadas al departamento. Si se requiere acceder por módems deberá ser a través de una "equipo puente", el cual es un equipo que tiene abiertos lo servicios para proveedores de Internet como módems de la UNAM y *Prodigy* así como aquellas computadoras dentro de Red UNAM.

Políticas sobre uso de las cuentas y contraseñas

- Es responsabilidad de los administradores cambiar contraseña del superusuario (*root*) y de las cuentas de los administradores cada mes, dos meses o según convenga. Si alguno de los administradores dejara de laborar en el Departamento de Administración de Servidores, modificar inmediatamente el password y notificar a los demás administradores.
- Las contraseñas y por consecuencia las cuentas de los administradores en un servidor de producción, serán bloqueados inmediatamente a la salida del miembro.
- El administrador deberá entrar siempre como usuario común y no como superusuario, al menos que se requiera entrar bajo circunstancias especiales, como no tener cuenta dentro de staff o bien cuando ocurre alguna emergencia que requiere entrar directamente como *root*.
- Las cuentas de usuario común de los administradores tendrán un nombre de grupo igual al del login, deberán cumplir con las características de cuentas únicas y letras

minúsculas. En cuanto a su uso es personal e intransferible, por lo cual no se permite compartir su cuenta ni su contraseña con persona alguna, aún si ésta acredita la confianza del usuario.

- Todos las contraseñas, tanto la de *root* como la de usuarios comunes deberán cumplir con las siguientes características:

- Contener caracteres alfanuméricos.
- Incluir algunos caracteres especiales($,%,&,!).
- Tener tanto mayúsculas como minúsculas.
- Contener palabras fáciles de recordar para el dueño pero no para intrusos, de este modo no necesitan escribirlos en papel.
- No deben ser palabras que estén dentro de un diccionario, secuencias conocidas de caracteres, datos personales ni acrónimos.
- No enviar password a través de e-mail
- Contener de 6 a 8 caracteres.

- Algunas precauciones para las contraseñas son las siguientes:

- No dejar la contraseña en lugares que sean fáciles de obtener (papel, cuaderno, pegada a la terminal, teclado o cualquier otra parte del sistema).
- Al escribir la contraseña no hacerlo de manera explicita combinar la contraseña con otras letras o palabras (incluso de la misma contraseña).
- No grabar la contraseña electrónicamente (archivo, base de datos, correo electrónico).
- No usar la contraseña de la cuenta como contraseña para otras aplicaciones por ejemplo, como juegos en línea, correos, etc.

Políticas sobre uso de Hardware y Software

- El administrador podrá hacer uso de programas que rastrean o explotan vulnerabilidades en los sistemas propios y no en los ajenos, con el fin verificar el

nivel de seguridad, se realizará un reporte detallado y se harán los cambios necesarios.

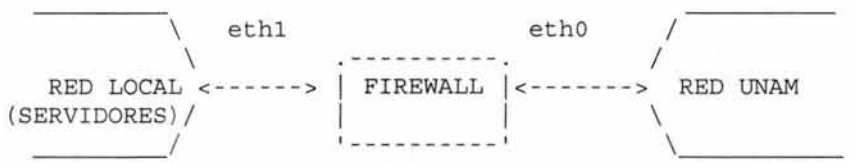
- El uso de herramientas para tener conocimiento de las características del servidor (como detección de puertos abiertos, vulnerabilidades o intento de ataque) en cualquiera de nuestro servidores deberá notificarse anticipadamente y por escrito. Si no se notifica por escrito de forma clara lo que se desea realizar, se podrá sancionar al responsable. Un mal empleo de herramientas en un servidor de producción puede causar inestabilidad en el mismo por ser usado en horas críticas. Esto aplica a los administradores y a usuarios externos.
- No se permitirá bajo ninguna circunstancia el uso de cualquiera de las computadoras con propósitos de ocio o lucro.
- El administrador deberá emplear regularmente herramientas de seguridad para verificar la integridad de los archivos del sistema, tales como: *Satan, Cops, John the ripper, Tripwire*, entre otras.
- Revisar que no existan archivos de *root* con permisos de escritura para todo el mundo.
- Revisar que no existan archivos *.rhosts* con permisos de escritura para todo el mundo.
- Revisar que los directorios del sistema (como /, /bin, /usr, /usr/bin, etc.) pertenezcan a *root* y a los grupo *sys* y other, según sea el caso.
- Revisar que no haya archivos cuyo dueño no aparezca en /etc/passwd.
- Revisar que los archivos de dispositivos y configuración tengan los permisos correctos.
- Revisar los usuarios con UID 0, investigar el por qué y cómo lo utilizan, y luego modificarlas.
- Revisar detalladamente los archivos ocultos, con espacio, con nombres como: *crack, irc, sniff, scan, .forward, shadow, passwd, hack y bitch.*
- El administrador deberá verificar que cada cuenta de los usuarios tenga contraseña y sea segura. Para ello hay que revisar periódicamente los archivos /etc/passwd y /etc/shadow ejecutando periódicamente el programa de John the ripper.

- El administrador deberá monitorear los recursos del servidor, memoria swap, tipo y número de procesos, espacio en disco, conexiones, entre otras.
- Cada administrador deberá realizar una bitácora de las actividades realizadas por cada servidor a su cargo (control de cambios). Entendiéndose por actividades como: cambio en la configuración, actualizaciones, alta y baja de usuarios; en general, todo aquello que se considere importante. Se indicará la fecha y tipo de acción por servidor. Las bitácoras deberán estar en un lugar seguro, por tanto no estarán almacenadas en el mismo sistema. Deberá estar de forma escrita como en un medio electrónico.
- Los *scripts* y bitácoras de las herramientas y aplicaciones referentes a la administración (estarán en cualquier de las siguientes rutas: /home/log, /home/logs o /home/root) ubicadas dentro de un File *System* propio o específico.
- Cada uno de los servidores en producción deberá tener mínimamente documentado:
  - Configuración de Hardware y Software.
  - Procedimientos.
  - Bitácoras.
  - Alarmas (clasificación y descripción).
  - La documentación deberá ser detallada y actual.
- El acceso al *site* será únicamente para los administradores o bien por personas autorizadas previamente por el jefe del departamento y acompañadas por al menos un administrador.
- El *site* deberá permanecer cerrado bajo llave por el administrador.
- El administrador cuidará que los conectores, cables y dispositivos periféricos no estén dispersos, mal acomodados o cualquier factor que pueda provocar algún daño o mal funcionamiento.
- El administrador se asegurará de mantener limpio el *site* periódicamente, aclarando que la limpieza se realizará por todos los administradores del Departamento.
- Cualquier aplicación, programa, herramienta o desarrollo que cause problemas de desempeño, ya sea de uso de memoria y/o procesador en el servidor, será

bloqueado temporalmente hasta que sea arreglado por el usuario o bien borrado si no lo resuelve.

REGLAS DEL FIREWALL

Ahora veremos reglas que se probaron para el Firewall de prueba en el Departamento, tomando el siguiente esquema red red de forma general.



Los siguiente se puede colocar en un *script*:

```
=-=-=-=-=-=-=-=-=-=
firewall.sh
=-=-=-=-=-=-=-=-=-=

REDLOCAL="172.16.0.0/24"
REDUNAM="132.248.0.0/16"
IPSPERM="132.248.120.96/24"


  .-------------------------------------.
  | CARGA DE MODULOS                    |
  '-------------------------------------'

/sbin/modprobe ip_tables
/sbin/modprobe ip_nat_ftp
/sbin/modprobe ip_conntrack
/sbin/modprobe ip_conntrack_ftp
```

```
.--------------------------------------.
| POLITICAS GENERALES |
'--------------------------------------'

iptables -F
iptables -P INPUT   DROP
iptables -P FORWARD DROP
iptables -P OUTPUT  DROP


.-----------------------------------------------------------------------------------------------------.
| RESTRICCIONES SOBRE ACCESOS Y SERVICIOS EN EL PROPIO FIREWALL |
'-----------------------------------------------------------------------------------------------------'

=-=-=-=-= LOOPBACK =-=-=-=-=

/sbin/iptables -A INPUT  -i lo -j ACCEPT
/sbin/iptables -A OUTPUT -o lo -j ACCEPT


=-=-=-=-= DNS DESDE EL FIREWALL HACIA INTERNET =-=-=-=-=

/sbin/iptables -A INPUT  -i eth0  -p udp --sport 53 -j ACCEPT
/sbin/iptables -A OUTPUT -o eth0  -p udp --dport 53 -j ACCEPT


=-=-=-=-= REGLAS NAT PARA ENMASCARAMIENTO =-=-=-=-=

/sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
/sbin/iptables -t nat -A POSTROUTING -o eth0 -s 172.16.0.0/24 -j MASQUERADE
=-=-=-=-= ACCESO AL FIREWALL POR SECURE SHELL =-=-=-=-=

/sbin/iptables -A INPUT  -i eth0 -s servidor -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j
ACCEPT
/sbin/iptables -A OUTPUT -o eth0 -d servidor -p tcp --sport 22 -m state --state ESTABLISHED    -j
ACCEPT


=-=-=-=-= PING DESDE EL FIREWALL =-=-=-=-=

/sbin/iptables -A INPUT -p icmp  -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A OUTPUT  -p icmp -m state --state NEW,ESTABLISHED    -j ACCEPT
=-=-=-=-=  TRACEROUTE DESDE EL FIREWALL HACIA INTERNET =-=-=-=-=

/sbin/iptables -A OUTPUT -o eth1 -p udp --sport 1024:65535 --dport 33434:33523 -m state --state NEW -j
ACCEPT


=-=-=-=-= PUERTOS PRIVILEGIADOS CERRADOS =-=-=-=-=

/sbin/iptables -A INPUT -p tcp -s 0/0 -d 0/0 --dport 0:1023 -j DROP
```

```
=-=-=-=-=  LOGGING DE TODO LO DEMAS =-=-=-=-=

/sbin/iptables -A INPUT  -j LOG --log-prefix "FIREWALL:REJECT_INPUT"  --log-level info
/sbin/iptables -A OUTPUT -j LOG --log-prefix "FIREWALL:REJECT_OUTPUT" --log-level info


 .-----------------------------------------------------------.
 |  POLITICAS DE ACCESO A INTERNET                           |
 '-----------------------------------------------------------'

=-=-=-=-=  CONNECTION TRACKING PARA CONEXIONES ORIGINADAS LOCALMENTE =-=-=-=-=

/sbin/iptables -A FORWARD -i eth1 -o eth0 -s ! $REDLOCAL -d   $REDLOCAL -m state --state
ESTABLISHED,RELATED -j ACCEPT


=-=-=-=-=  DNS Y HTTP  =-=-=-=-=

/sbin/iptables -A FORWARD -i eth1 -o eth0 -s   $REDLOCAL -d ! $REDLOCAL -p udp --dport 53  -j
ACCEPT
/sbin/iptables -A FORWARD -i eth1 -o eth0 -s   $REDLOCAL -d ! $REDLOCAL -p tcp --dport 53  -j
ACCEPT
/sbin/iptables -A FORWARD -i eth1 -o eth0 -s   $REDLOCAL -d ! $REDLOCAL -p tcp --dport 80  -j
ACCEPT


 .-----------------------------------------------------------------------------------------.
 |    POLITICAS DE SERVICIOS DISPONIBLES EN REDLOCAL                                       |
 '-----------------------------------------------------------------------------------------'

=-=-=-=-=  CONNECTION TRACKING PARA CONEXIONES ORIGINADAS DESDE INTERNET =-=-=-=-
=

/sbin/iptables -A FORWARD -i eth0 -o eth1 -s  $REDLOCAL -d ! $REDLOCAL -m state --state
ESTABLISHED -j ACCEPT


=-=-=-=-=  SECURE SHELL =-=-=-=-=

/sbin/iptables -A FORWARD -i eth1 -o eth0 -s   $IPSPERM  -d  $REDLOCAL   -p tcp --dport 22  -j
ACCEPT
/sbin/iptables -A FORWARD -i eth1 -o eth0 -s   $IPSPERM -d  $REDLOCAL   -p tcp --dport 22  -j
ACCEPT


=-=-=-=-=  LOGGING DE TODO LO DEMAS =-=-=-=-=

/sbin/iptables  -A FORWARD -j LOG --log-prefix "FIREWALL:REJECT-FORWARD " --log-level info
/sbin/iptables -A FORWARD -m tcp -p tcp -j LOG
/sbin/iptables -A FORWARD -m udp -p udp -j LOG
/sbin/iptables -A FORWARD -m udp -p icmp -j LOG
```

=-=-=-=-= JUEGOS, CHAT, MUSICA RECHAZADOS =-=-=-=-=

Red de Audio Galaxy
/sbin/iptables -A FORWARD -d 64.245.58.0/23 -j REJECT

GNUtella, Bearshare y ToadNode
/sbin/iptables -A FORWARD -p TCP --dport 6346 -j REJECT
Puertos y redes de Kazaa y Morpheus
/sbin/iptables -A FORWARD --dport 1214 -j REJECT
/sbin/iptables -A FORWARD -d 213.248.112.0/24 -j REJECT
/sbin/iptables -A FORWARD -d 206.142.53.0/24 -j REJECT

Red de Napigator
/sbin/iptables -A FORWARD -d 209.25.178.0/24 -j REJECT

Red de Napster
/sbin/iptables -A FORWARD -d 64.124.41.0/24 -j REJECT

Redes de WinMX
/sbin/iptables -A FORWARD -d 209.61.186.0/24 -j REJECT
/sbin/iptables -A FORWARD -d 64.49.201.0/24 -j REJECT

Red de IMesh
/sbin/iptables -A FORWARD -d 216.35.208.0/24 -j REJECT
Programas de mensajeria

AIM e ICQ
/sbin/iptables -A FORWARD --dport 9898 -j REJECT
/sbin/iptables -A FORWARD --dport 5190:5193 -j REJECT
/sbin/iptables -A FORWARD -d login.oscar.aol.com -j REJECT
/sbin/iptables -A FORWARD -d login.icq.com -j REJECT

Jabber
/sbin/iptables -A FORWARD --dport 5222:5223 -j REJECT

MSN Messenger
/sbin/iptables -A FORWARD -p TCP --dport 1863 -j REJECT
/sbin/iptables -A FORWARD -d 64.4.13.0/24 -j REJECT

Yahoo! Messenger
/sbin/iptables -A FORWARD -p TCP --dport 5000:5010 -j REJECT
/sbin/iptables -A FORWARD -d cs.yahoo.com -j REJECT

SI SE DESEA IMPLEMENTAR ESTOS SERVICIOS LAS REGLAS QUE SERÍAN ÚTILES PODRÍAN SER COMO ESTAS:

=-=-=-=-= SMTP HACIA SERVIDOR MAIL =-=-=-=-=

/sbin/iptables -A INPUT -i eth1 -s servmail -p tcp --sport 25 -j ACCEPT
/sbin/iptables -A OUTPUT -o eth1 -d servmail -p tcp --dport 25 -j ACCEPT

=-=-=-=-= SMTP ENTRANTE =-=-=-=-=

/sbin/iptables -A FORWARD -i eth0 -o eth1 -s ! $REDLOCAL -d servmail -p tcp --dport 25 -j ACCEPT

=-=-=-=-= POP =-=-=-=-=

/sbin/iptables -A FORWARD -i eth0 -o eth1 -s 0/0 -d servmail -p tcp --dport 110 -j ACCEPT

=-=-=-=-= IMAP =-=-=-=-=

/sbin/iptables -A FORWARD -i eth0 -o eth1 -s 0/0 -d servmail -p tcp --dport 143 -j ACCEPT

=-=-=-=-= IMAP SSL =-=-=-=-=

/sbin/iptables -A FORWARD -i eth0 -o eth1 -s 0/0 -d servmail -p tcp --dport 110 -j ACCEPT

=-=-=-=-= HTTP =-=-=-=-=

/sbin/iptables -A FORWARD -i eth0 -o eth1 -s ! $REDLOCAL -d www -p tcp --dport 80 -j ACCEPT
/sbin/iptables -A FORWARD -i eth0 -o eth1 -s ! $REDLOCAL -d servwww -p tcp --dport 80 -j ACCEPT

GLOSARIO

ACCEPT regla de decisión de filtrado de Firewall para pasar el paquete a través de Firewall hasta el siguiente destino.

ACK indicador de TCP que confirma la recepción de un segmento de TCP recibido anteriormente

Ataque por denegación de servicio ataque basado en la idea del envío de datos inesperados o de inundar un sistema con paquetes para interrumpir o alentar seriamente su conexión a Internet, disminuyendo el rendimiento de los servidores hasta el extremo en que no es posible atender las peticiones legitimas, o en el peor de los casos, bloqueando todo el sistema.

Autenticación Proceso de determinar que una entidad es quien o lo que dice ser.

AUTH Puerto de servicio TCP 113, asociado con el servidor de autenticación de usuario identd.

Bug Problema que evita que un programa funcione de la manera apropiada.

CERT *Computer Emergency Response Team*, (Equipo informático de respuestas de emergencia), un centro de coordinación de información de seguridad de Internet creado en el instituto de ingeniería de software de la Universidad Carnegie Mellon después del incidente Internet Worm, en 1988.

Crack Programa de adivinación de contraseñas.

Cracker Palabra inglesa que significa "intruso". Se trata de un individuo que intenta penetrar en una computadora o sistema informático ilegalmente con intenciones perversas (robar información, borrarla, etc.).

Cron Sistema demonio llamado crond y unos archivos de configuración y secuencias de comandos que ponen en marcha tareas del sistema programadas.

Datagrama IP Paquete de la capa de red IP.

Demonio Servidor de servicios de sistema básicos que se ejecuta en segundo plano.

DMZ Zona desmilitarizada, perímetro de red que contiene máquinas que atienden servicios públicos, separados de la red privada local. Los servidores públicos menos seguros están aislados de la LAN privada.

DNS Domain Name System (Sistema de nombres de Domino); un servicio global de base de datos de Internet que permite inicialmente a los clientes la búsqueda de direcciones IP de host, dado el host completamente cualificado y el nombre de dominio, asi como para buscar nombres de equipo completamente cualificados, dadas sus direcciones IP.

Dirección IP identificador numérico único que se asigna a una red especifica o a una interfaz de red de un dispositivo específico en una red. Es una dirección software que se puede traducir

directamente a un host o nombre de red comprensible por el usuario. Las direcciones IP de interfaz de red de host también se asocian con una o más direcciones de interfaz de hardware.

Directiva aceptar todo de forma predeterminada directiva que acepta todos los paquetes que no coinciden con una regla de Firewall de la cadena. Por tanto, casi todas las reglas de Firewall son reglas DENY que definen las excepciones a la directiva aceptar predeterminada

Directiva denegar todo en forma predeterminada directiva que elimina todos los paquetes que no cumplen una regla del Firewall en la cadena. Por tanto, la mayoría de las reglas del Firewall son reglas ACCEPT que definen las excepciones a la directiva denegar predeterminada.

Directiva predeterminada directiva para un conjunto de reglas, ya sea para una cadena input, o para una cadena forward, que define una disposición del paquete cuando éste no coincide con ninguna regla del conjunto.

Enmascaramiento Proceso se sustituir una dirección origen local de un paquete saliente con la del Firewall o máquina que hace de pasarela, de forma que permanezcan ocultas las direcciones IP de la LAN. El paquete parece proceder de la máquina de pasarela en lugar de una máquina interna de la LAN. El proceso se invierte para paquetes de respuesta entrantes desde servidores remotos. La dirección de destino del paquete, la dirección IP de la máquina Firewall, se sustituye con la dirección de la máquina cliente dentro de la LAN interna. El enmascaramiento IP de suele llamar traducción de direcciones red (NAT, network addres translation).

Ethernet Especificación de LAN de banda base, inventada por lo corporación Xerox desarrollada en conjunto por Xerox, Intel y Digital Equipment Corporation. Las redes Ethernet utilizan métodos de acceso CSMA/CD y corren sobre una gran variedad de tipos de cables a 10 Mbps. La red Ethernet es similar a los estándares de la serie IEEE 8002.3.

Filtro, Firewall regla de filtrado de paquetes de un Firewall, o paquete de exploración, que define las características del paquete. Si coinciden, determina si se permite que el paquete pase a través dela interfaz de red o se elimina. Los filtros se definen en términos de un origen de paquetes y direcciones destino, puertos destino y origen, tipo de protocolo, estado de la conexión TCP y tipo de mensaje ICMP.

Firewall Firewall o cortafuegos es un sistema o grupo de sistemas que hace cumplir una política de control de acceso entre dos redes. De una forma más clara, podemos definir un Firewall como cualquier sistema (desde un simple router hasta varias redes en serie) utilizado para separar en cuanto a seguridad se refiere una máquina o subred del resto, protegiéndola así de servicios y protocolos que desde el exterior puedan suponer una amenaza a la seguridad. El espacio protegido, denominado perímetro de seguridad, suele ser propiedad de la misma organización, y la protección se realiza contra una red externa, no confiable, llamada zona de riesgo.

Firewall, bastión Un Firewall que tiene dos o más interfaces de red y es la pasarela o punto de conexión entre esas dos redes, la mayoría entre una LAN e Internet. Como un Firewall bastión es un único punto de conexión entre redes, el bastión se asegura con todas las medidas posibles.

Firewall, filtrado de paquetes Firewall que se implementa en la red y en las capas de transporte que filtran el tráfico de red en función de los paquetes, tomando decisiones de enrutamiento basándose en la información del encabezado del paquete IP.

GNU Nombre con el que se conoce a toda aplicación compatible con UNIX que es totalmente gratuita. Proyecto iniciado por Richard Stallman en 1983 en el Instituto de Tecnología en Massachutets.

HOWTO Además de las páginas man estándar, Linux incluye documentación interactiva sobre muchos temas, en muchos idiomas y en varios formatos. Los documentos HOWTO se coordinan y mantienen por el proyecto de documentación de Linux.

ICANN (Internet Corporación for Asigned Names and Numbers) Lo más parecido a una "autoridad central" en Internet es tan solo un órgano de gestión administrativa que no puede impedir a ninguna persona o institución el acceso a Internet ni legal ni técnicamente.

ICMP Internet Contros Message Protocol (Protocolo de control de mensajes Internet); un estado IP de la capa de red y un mensaje de control.

IMAP Internet Message Access Protocol (Protocolo de Acceso de Mensajes Internet); se usa para recuperar correo de servidores de correo que ejecutan un servidor IMAP.

inetd.conf Archivo de configuración de inetd.

Internet Nombre con el que se conoce a una agrupación de redes informáticas interconectadas de todo el mundo que permiten la comunicación entre millones de usuarios de todo el planeta. La intención original de su creación fue la de conectar las universidades y centros de investigación de todo el mundo, aunque se ha convertido en el principal medio de comunicación de usuarios, empresas y todo tipo de organizaciones.

IRC *Internal Relay Chat* Transmisión de conversación por Internet.

LAN Local Area Network (Red de área local).

Localhost Nombre simbólico que se suele utilizar para la interfaz de bucle invertido de una maquina en /etc/hosts.

Memoria swap Parte reservada del disco que el *kernel* usa durante el procesamiento de forma temporal.

Modelo de referencia OSI (Open System Interconnection, Interconexión de sistemas abiertos) Un modelo de siete capas de la organización internacional para la estandarización (International Organization for Standardization) que marca una línea de trabajo o directrices para los estándares de interconexión de redes.

Modelo de referencia TCP/IP Modelo de comunicación de red informal de desarrollo cuando, TCP/IP se convirtió en el estándar de facto para las comunicaciones de Internet entre las máquinas UNIX a finales de los años 70´s y comienzos de los años 80´s. En lugar de ser un ideal académico y formal, el modelo de referencia TCP/IP se basa en lo que los fabricantes y programadores se pusieron de acuerdo que debería ser la comunicación en Internet.

MTU Maximun Transmission Unit (Unidad de transmisión máxima); el máximo tamaño de paquete en función de la red subyacente.

NAT Network Address Translation, traductor de direcciones de red, proceso de traducción de dirección IP no homologadas a direcciones IP homologadas, o que son validas para Internet y viceversa.

Netstat Un programa que registra distintos tipos de estados de red en función de varias tablas de núcleo relacionadas con la red.

Pasarela Un equipo o programa que sirve como conducto o transmisión, entre dos redes.

Ping Una herramienta sencilla de análisis de red que se usa para determinar si es posible contactar con un host remoto y esta preparado para responder. *Ping* envía un mensaje de solicitud de eco ICMP. El host receptor devuelve un mensaje de respuesta de eco ICMP.

POP Post Office Protocol (Protocolo de oficina de correos); se usa para recuperar el correo de los host que ejecutan un servidor POP.

Proxy Un programa que crea y mantiene una conexión de red en beneficio de otro programa. Proporcionando un conducto en el nivel aplicación entre un cliente y un servidor. El cliente y el servidor, en realidad, no tienen comunicación directa. El proxy aparenta ser el servidor para el programa cliente y parece ser el cliente para el programa servidor.

Proxy, pasarela de aplicación Parecida a un Firewall de exploración de host que se implementa en la configuración del sistema y en los niveles de aplicación. Solo el equipo que ejecuta la pasarela de aplicación tiene acceso directo a Internet. El tráfico de red nunca se envía automáticamente a través de la pasarela de aplicación. Todo el acceso a Internet se realiza a través del programa de pasarela. Solo se permite el acceso externo hacia la máquina de pasarela . El acceso interno solo se permite hacia la maquina de pasarela. Los usuarios locales deben iniciar una sesión de usuario en la maquina de pasarela y tener acceso a Internet desde ahí, o conectar con la pasarela de aplicación y autenticarse primero a sí mismos. Un servidor proxy se suele implementar como una aplicación independiente para cada servicio que hace uso del proxy. El servidor proxy del nivel aplicación entiende el protocolo de comunicación específico de la aplicación. Cada aplicación proxy aparenta ser el servidor para el programa cliente y aparenta ser el cliente para el servidor real. Los programas cliente especiales, o los programas cliente configurados especialmente, se conectan a un servidor proxy en lugar de un servidor remoto. El proxy establece la conexión con el servidor remoto en beneficio de la aplicación cliente, después de sustituir la dirección de origen del cliente con la suya.

Proxy, pasarela de circuito Un servidor proxy que se puede implementar como aplicaciones separadas para que cada servicio pueda hacer uso del proxy, como una transmisión generalizada de la conexión que no tiene un conocimiento específico sobre protocolos de aplicación. La transmisión a nivel de circuito crea un circuito de conexiones virtuales administradas por software entre un cliente y un programa servidor. Al contrario que los servidores proxy a nivel de aplicación, los pasos intermedios de la conexión se realizan de forma transparente hacia el usuario. SOCKS es un sistema proxy a nivel de circuito.

Puerto En TCP o UDP, el designador numérico de un canal particular de comunicación de red. Las asignaciones de puerto las administra el IANA. Algunos puertos se asignan a protocolos de comunicación de una aplicación particular como parte del estándar del protocolo. Algunos

puertos se registran como asociados con un servicio particular por convención. Algunos puertos tienen libertad de asignación para que los unen los clientes y los protocolos de usuario.

Privilegiado puerto del intervalo que va desde 0 hasta 1023. Muchos de estos puertos los asigna el estándar internacional a protocolos de aplicación. El acceso a puertos privilegiados requiere privilegios del nivel del sistema.

No privilegiados puerto del intervalo que va desde 1024 hasta 65535. Algunos de estos puertos se registran y se usan según una convención para ciertos programas. Cualquier puerto que pertenezca a este intervalo lo puede utilizar un programa cliente que establecer una conexión con un servidor de la red.

REJECT Decisión de regla de filtrado Firewall para eliminar un paquete y devolver al emisor un mensaje de error ICMP.

RELAYING Acción de hacer uso de un servidor como medio de difusión de correo electrónico en el que ni el remitente ni el destinatario son usuarios de dicho servidor.

RFC *Request for Comment* (Peticiones de comentarios); una nota de recordatorio publicada a taves de Internet Society (Sociedad de Internet) o del Internet Engineering Task Force (Grupo de Trabajo de Ingeniería de Internet). Algunos RFC se convierten en estándares. Los RFC suelen estar relacionados con temas de Internet o el conjunto de protocolos TCP/IP.

Ruteador o router Dispositivo de la capa de red que utiliza una o más medidas para determinar la trayectoria óptima a lo largo de la cual deba diseccionarse el tráfico de la red. Los ruteadores direccionan paquetes de una red a otra con base en la información de la capa de red, también se refieren a ellos como compuertas.

RPC Remote-procedure call (llamada a procedimiento remoto).

SATAN Security Administrator Tool for Analyzing Networks (herramienta del administrador de seguridad para análisis de redes); una herramienta que ayuda a identificar posibles debilidades de seguridad en configuraciones de servicios de red.

Segmento, TCP Mensaje TCP

Servidor Genéricamente, dispositivo de un sistema que resuelve las peticiones de otros elementos del sistema, denominados clientes. Computadora que suministra servicios a los usuarios de la red, el servidor recibe solicitudes para los diferentes servicios y administra las solicitudes para que sean atendidas de una forma ordenada y en secuencia.

Software Programa de sistemas, utilerías o aplicaciones expresadas en un lenguaje legible para las computadoras.

Spam Palabra inglesa que se usa para indicar el hecho de recibir gran cantidad de mensajes propagandísticos vía correo electrónico.

Setgid Un programa que, cuando se ejecuta, asume el Id del grupo del propietario el programa, en lugar de la Id del grupo del proceso que ejecuta el programa.

Shell Un interprete de comandos UNIX, como sh, ksh, bash y csh.

SYN Indicador de solicitud de sincronización TCP. Un mensaje SYN es el primer mensaje que envía un programa para abrir una conexión con otros programas presentes en la red.

Syslog.conf Archivo de configuración del demonio de inicio de sesión.

Syslogd El demonio de inicio de sesión del sistema, que recopila mensajes de estado y error generados por los programas que envían mensajes mediante la llamada al sistema syslog().

TCP Transmission Control Protocol (Protocolo de control de la transmisión); se usa para conexiones de red activas y fiables entre dos programas.

Token ring LAN con protocolo de acceso de estafeta circulante desarrollado y soportado por IBM. La red de Token ring corre a 4 o 16 MBPS sobre una topología de anillo. Es similar al estándar IEEE 802.5.

Traceroute Herramienta de análisis de red que se usa para determinar la ruta de un equipo a otro a través de la red.

UDP User Datagram Protocol (Protocolo de datagrama de usuario); se usa para enviar mensajes de red individuales entre programas sin ninguna garantía de envío u orden de envío.

BIBLIOGRAFÍA

BRENT, Chapman, ZWICKY Elizabeth. "*Construya Firewalls para Internet*", O'Reilly, México, *1997.*

ZIEGLER, Robert L. *"Linux Firewalls* ", New Riders professional library,2000.

SIYAN, Karanjit. *"Firewalls y la seguridad en Internet"*, prentice hall, 1997.

CHESWICK, William, BELLOVIN Steven. *"Firewalls and Internet Security: Repelling the Wily Hacker"*, Addison-Wesley, 1994.

GARFINKEL, Simson. *Seguridad Práctica en UNIX e Internet*, O'Reilly, México, 1999.

MARTÍNEZ IBÁÑEZ, Jesús, GÓMEZ SKARMETA Antonio, Martínez Barberá, Humberto. *"Seguridad en Internet: Ataques, Técnicas y Firewalls"*, Novática, nº 127, pp. 23-30, 1997.

REFERENCIAS DE INTERNET

BLFS , Equipo de Desarrollo. *"Más Allá de Linux From Scratch"*, Proyecto LFS, 2001-2003, http://www.escomposlinux.org/lfs-es/blfs-es-CVS/index.html, consultada en 2003.

"redhat Linux 8.0The Official Red Hat Linux x86 Installation Guide", Red hat inc, sección Firewall,2001,http://www.redhat.com/docs/manuals/linux/RHL-7.1-anual/install-guide/s1-steps-install-cdrom.html#S2-STEPS-MAKE-DISKS, consultada en 2003.

"Más Allá de Linux From Scratch", Proyecto LFS, Capítulo 4. Seguridad,2003, http://www.escomposlinux.org/lfs-es/blfs-e|-CVS/postlfs/iptables.html, consultada en 2003.

NETFILTER. "*Packet Filtering HOWTO* ", 1999-2004, secciones Networking Concepts, HOWTO, NAT HOWTO, consultada en 2003.

"ISS Wireless FAQ", sección FAQ, http://www.iss.net/wireless/

"Seguridad en Ordenadores y Redes",edicion Internet, http://jo.morales0002.eresmas.net/fsegur.html, consultada en 2003.

KIRCH Olaf, DAWSON Terry. *"Guía de Administración de Redes con Linux"*,O'Reilly (printed version) (c) 2000 O'Reilly & Associates,Proyecto LuCAS por la traducción al español HispaLiNUX, http://lucas.hispalinux.es/Manuales-LuCAS/GARL2/garl2/index.html, consultada en 2003.

***"Redes en linux Como"*,Universidad Nacional de Salta CONICET, tema 6Información relacionada con IP y Ethernet, http://g.unsa.edu.ar/doc/howto/es/html/Redes-En-Linux-Como.html#toc6**, consultada en 2003.

"El módulo Netfilter de Linux, iptables", http://www.redes.upv.es/irc/trabajos/IR-iptables.pdf, consultada en 2003.

"Interconexión de redes", 2003, http/www.redes.upv.es/irc/examenes/notassep.html, consultada en 2003.

"COMO del cortafuegos (firewall)", http://linuxcol.uniandes.edu.co/~gramo/comos_gramo/UMAN.2001-1/firewall-howto/, consultada en 2003.

VILLALÓN, Antonio. *"Seguridad en Unix y Redes"*, Open Publication License, 2002, http://www.hispasec.com/unaaldia.asp, consultada en 2003.

VILLALÓN, Antonio. Seguridad en UNIX y Redes", http://andercheran.aiind.upv.es/toni/personal/unixsec.pdf, consultada en 2003.

"Firewalls", IEEE USM Student Branch, ttp://www.ieee.utfsm.cl/manuales/firewalls1.pdf, consultada en 2003.

"Evaluación de las distintas soluciones firewalls", http://www-mat.upc.es/~jforne/firewalls.pdf, consultada en 2003.

"Visión General del Firewall Raptor", ttp://www.insys-corp.com.mx/newinsys/Productos/RaptorFW/Default.htm, consultada en 2003.

NEMO. *"Gestión de firewalls: una metodología de gestión"*, 2001, http://www.deepzone.org/editions/others/gestion.pdf , consultada en 2003.

"Redes Virtuales VPN",
http//atenea.udistrital.edu.co/egresados/xsepulveda/temas/hernan/VPN.htm", consultada en 2003.

"Control de Accesos", edición de Internet,
http://ants.dif.um.es/~humberto/asignaturas/v30/docs/Accesos.pdf, consultada en 2003.

GURUTZE, Marijuán. *"Ponencias Internet 99"*, bilbomática, 1999, http://www.aui.es/biblio/libros/mi99/5seguridad_integral.htm, consultada en 2003.

CARVAJAL, Roberto. *" FIREWALL Y NAT en Linux", Grupo Linux UNAB"*, 2002, http://fis.unab.edu.co/docentes/rcarvaja/publicaciones/Firewall.htm, consultada en 2003.

"Seguridad en Redes", arcert, 1999, www.arcert.gov.ar/webs/manual/manual_de_seguridad.pdf+lsof+manual&hl=es&lr=lang_es&ie=UTF-8, consultada en 2003.

Grennan, Mark. *"Cómo de cortafuegos y servidor proxy"*, 2001,
http://www.linux-es.com/docs/HOWTO/translations/es/Cortafuegos-COMO, consultada en 2003.

'Instalación de iptables", http://www.escomposlinux.org/lfs-es/blfs-es-CVS/postlfs/iptables.html, consultada en 2003.

PARDO, Juan. *"Instalación linux"*, http://www.juanon14.8m.com/trabajos/instso/instalac.htm, consultada en 2003.

GONZÁLEZ, Jorge. *"Activación del firewall (IPTABLES)"*, http://debaser.ath.cx/jorgegv/como-montar-nodo/fw-ident-ints.html, consultada en 2003.

CURSOS

ZARAGOZA, Fernando, "Seguridad en Unix", DGSCA, UNAM, México D.F., 2002 http://newman.posgrado.unam.mx/ds/cursos/seguridad , sección "Políticas de seguridad"

RIVERA M., Hugo. *"Ruteó en redes de datos"*, DGSCA, UNAM, México D.F., 2003.

"Instalación y Administración Linux" DGSCA, UNAM, México D.F., 2003.