1

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN

REDES DE COMPUTADORAS. SEGURIDAD EN INTERNET. IMPLEMENTACION DE UN FIREWALL EN FES CUAUTITLAN.

TRABAJO DE SEMINARIO

QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN INFORMATICA

P R E S E N T A :

JUAN GABRIEL ARENAS ROSAS

ASESOR: ING. JESUS MOISES HERNANDEZ DUARTE.

CUAUTITLAN IZCALLI, ESTADO DE MEXICO. 200

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN

UNIDAD DE LA ADMINISTRACION ESCOLAR

DEPARTAMENTO DE EXAMENES PROFESIONALES

U. N. A. M.
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLAN

DEPARTAMENTO DE
EXAMENES PROFESIONALES

DR. JUAN ANTONIO MONTARAZ CRESPO
DIRECTOR DE LA FES CUAUTITLAN
P R E S E N T E

ATN: Q. Ma. del Carmen García Mijares
Jefe del Departamento de Exámenes
Profesionales de la FES Cuautitlán

Con base en el art. 51 del Reglamento de Exámenes Profesionales de la FES-Cuautitlán, nos permitimos comunicar a usted que revisamos el Trabajo de Seminario:

Redes de computadoras. Seguridad en Internet. Implementación de un Firewall en FES Cuautitlán.

que presenta el pasante: Juan Gabriel Arenas Rosas

con número de cuenta: 9201117-1 para obtener el título de :

Licenciado en Informática.

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXÁMEN PROFESIONAL correspondiente, otorgamos nuestro VISTO BUENO.

A T E N T A M E N T E
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautitlán Izcalli, Méx. a 30 de Octubre de 2000.

MODULO	PROFESOR	FIRMA
I	Ing. Jesús Moisés Hernández Duarte	
II	Ing. Carlos Vazquez Cruz	
IV	M.C.C. Araceli Nivón Zaghi.	

AGRADECIMIENTOS

A mis padres:

Con todo mi sincero y más profundo agradecimiento,
les dedico este trabajo a mis padres, que con su todo duro trabajo
y confianza me han apoyado en todas mis metas en la vida,
y que gracias a su consejo y enseñanzas he podido realizar uno
de mis más grandes sueños y el de ellos también

A mis Hermanos:

Que siempre han creído en mi y que de alguna u otra forma me
han apoyado, esto también es de ustedes, y que esto te motive Gina
para que sigas adelante con tus metas.

A la familia:

Me siento orgulloso de contar con una familia tan unida, con tíos y tías
Que han puesto su confianza en mí y que yo se que se sienten orgullosos.
Gracias.

Al Ing. Martín Alvarez Torres:

Gracias a usted Martín por darme la primera oportunidad de trabajar en
Este campo y de siempre motivarme y ayudarme para seguir adelante y estar al tanto
De mi desarrollo académico.

Para Magdalena:

A ti flaca por apoyarme y aguantar todo este tiempo y por recordarme cual es el punto de vista correcto que debo de tener, sigue con tus sueños.

SEGURIDAD EN INTERNET. IMPLEMENTACIÓN DE UN FIREWALL EN FES CUAUTITLÁN

ÍNDICE

4. IMPLEMENTACIÓN DE UN FIREWALL PARA EN FES CUAUTITLÁN

OBJETIVO GENERAL

- Diseñar e implementar un Firewall para FES Cuautitlán, empleando las principales estrategias de seguridad disponibles en la actualidad

OBJETIVOS PARTICULARES

- Identificar los servicios de Internet mayormente utilizados y potencialmente peligrosos.

- Analizar las principales arquitecturas de seguridad diseñadas para construir Firewalls en sitios conectados a Internet.

INTRODUCCIÓN

El Internet es una colección de redes alrededor del mundo que usan un protocolo común de comunicaciones. Muchas organizaciones están en el proceso de conectarse a el Internet para tomar ventaja de los servicios y recursos que ofrece. Negocios y agencias y cualquier otra organizacion (educativa, gobierno, salud, etc.) ahora están usando Internet o están considerando accesar a esta red mundial para una variedad de propósitos, incluyendo servicios como intercambio de correo electrónico, distribución de información del negocio o de la organización al público. Muchas organizaciones están conectando sus redes LAN internas a el Internet teniendo como consecuencia que sus estaciones de trabajo una conexión directa a el Internet.

La conexión a Internet puede ofrecer enormes ventajas, sin embargo, existe un aspecto que necesita mayor atención y este es el de la seguridad cuando se planea tener una conexión a el Internet. Existen significativos riesgos de seguridad asociados con Internet que con frecuencia no son obvios para los usuarios nuevos y aún para los existentes. En particular, existen personas que se dedican a planear intrusiones en algún sitio, aprovechándose de los puntos vulnerables de los sitios que estan conectados a Internet.

Las actividades de intrusión son difíciles de predecir y en ocasiones pueden ser difíciles de descubrir y corregir. Muchas organizaciones ya han perdido mucho tiempo y dinero en estar vigilando las actividades de intrusión, y lo que es más algunas organizaciones han sufrido daños en sun reputación como resultado de estas actividades.

El presente trabajo se enfoca especificamente en aspectos de seguridad para la organización que ya esta conectada a Internet o están por conectarse. En particular, este trabajo se enfoca en Firewalls para Internet como uno de los mecanismos y métodos usados para proteger sitios contra amenazas.

Nos centraremos en construir un Firewall en FES Cuautitlan, aplicando principalmente las técnicas y procedimientos analizados en este trabajo.

En el capitulo I abordaremos y centraremos la atención en Internet tocando puntos básicos para entender que es Internet pasando por los servicios que ofrece Internet (FTP, Telnet, correo electrónico), así como tocando aspectos de seguridad que comúnmente se presentan al conectarse a Internet.

En el capitulo II nos adentraremos en los Firewalls, cubriendo tópicos básicos, tales como, concepto de Firewalls, problemas con Firewalls adémas de tratar los componentes de un Firewall.

En el capitulo III abordaremos las principales técnicas y procedimientos usados actualmente para construir una solucion de Firewall, tocando temas como: filtrado de paquetes, sistemas proxy, entre otros.

En el capitulo IV se integrará el diseño del Firewall propuesto para la FES Cuautitlán.

Afortunadamente, existen soluciones disponibles que pueden ser usadas para mejorar la seguridad en un sitio. Un sistema de Firewall es una tecnica que ha probado ser altamente efectiva para mejorar los niveles de seguridad. Un Firewall es una colección de sistemas, routers y políticas puestos en un sitio central de conexión a la red. Un Firewall forza a todas las conexiones que pasan a traves de un gateway o router a que sean examinados y evaluados antes de ejecutar alguna acción peligrosa de acuerdo a las políticas de seguridad.

Un Firewall bien configurado puede actuar como un vehículo de relaciones públicas para la organización y puede ayudar a presentar una imagen favorable de la organización a el mundo entero.

CAPÍTULO 1

INTRODUCCIÓN AL INTERNET Y CONCEPTOS DE SEGURIDAD.

1.1 EL INTERNET

El Internet es una "red de redes" de nivel mundial que usa el protocolo TCP/IP para comunicaciones. El Internet fue creado inicialmente para ayudar a las comunicaciones entre investigadores auspiciados por el gobierno. A través de los 80´s, el Internet creció rápidamente incluyendo instituciones educativas, agencias de gobierno, organizaciones comerciales e organizaciones internacionales. En los 90´s, el Internet ha tenido un crecimiento fenomenal, con conexiones que se incrementan cada vez más rápido que cualquier otra red que haya sido creada (incluyendo la red de telefonía). Muchos millones de usuarios están ahora conectados a Internet.

1.2 INTERNET HOSTS

Muchos sistemas conectados a el Internet corren en diferentes versiones del Sistema Operativo UNIX. TCP/IP fue primeramente implementado a principios de los 80´s para la versión de UNIX escrita en la Universidad de Berkeley en California conocida también como Berkeley Software Distribution (BSD). Muchas versiones modernas de UNIX se derivan directamente del código de las versiones BSD, por lo tanto UNIX provee más o menos un conjunto estándar de servicios TCP/IP. Este estándar ha resultado en que diferentes versiones de UNIX hayan sufrido de las mismas vulnerabilidades, sin embargo, este también provee medios comunes para implementar estrategias de Firewall tales como Filtrado de paquetes.

Aunque UNIX es el Sistema Operativo predominante entre los Servidores de Internet, muchos otros tipos de sistemas y computadoras están conectadas a Internet, incluyendo sistemas como Digital Equipment Corporation´s VMs, NeXT, sistemas operativos de mainframe, y sistemas operativos de pc tales como Dos, Microsoft Windows, y sistemas Apple. Aunque los sistemas de ps con frecuencia ofrecen servicios de cliente, se puede usar Telnet para conectarse. Mientras que es bueno que la mayoría de los recursos de la red estén disponibles, este también tiene negativas consecuencias.

1.3 ESTRATEGIAS DE SEGURIDAD

1.3.1 MENOR PRIVILEGIO

Quizás el principio de seguridad más fundamental (Cualquier tipo de seguridad no sólo la de computadoras y redes) es el menor privilegio. Básicamente, el principio de menor privilegio, significa que cualquier objeto (usuario, administrador, programa, sistema o lo que sea) debe tener sólo los privilegios para cumplir con sus tareas asignadas (no más). Menor privilegio es un principio importante para limitar su exposición a tanques y para limitar el daño causado por ataques específicos.

Algunos fabricantes de automóviles ponen seguros para que una llave funcione en las puertas y el encendido, y otra en la guantera y la cajuela; de esa forma. puede cumplir con el menor privilegio al darle a la persona que le atiende en un estacionamiento la habilidad de estacionar el automóvil sin que tenga acceso a las cosas guardados en la cajuela: Muchas personas usan llaveros que se separan por la misma razón. Puede cumplir con el menor privilegio al darle a alguien la llave de su automóvil sin darle también la llave de su casa.

En el contexto de Internet, los ejemplos son interminables: Cada usuario tal vez no necesita tener acceso a cada servicio de esa red. Cada usuario no necesita modificar (o incluso leer) cada archivo de su sistema. Cada usuario quizá no necesita saber la contraseña del usuario root de la máquina. Cada administrador del sistemas probablemente no necesita saber las contraseñas de error de todos los sistemas. Cada sistema tal vez no necesita acceder a todos los archivos de cada sistema.

Aplicar el principio de menor privilegio sugiere que debe explorar formas de reducir privilegios necesarios para hacer varias operaciones. Por ejemplo:

1. No le de a un usuario la contraseña de root para un sistema si todo lo que necesita es reinstalar el servicio de impresión en lugar de eso, escriba un programa con los privilegios suficientes para que el usuario pueda reinstalarlo.

2. No haga que un programa ejecute setuid como root si lo único que necesita es escribir a un archivo protegido. En lugar de eso, haga que el archivo pueda ser escrito por algún grupo y haga que el programa ejecute setgid a ese grupo en lugar de setuid como root.

3. No haga que sus sistemas internos confíen en sus máquinas de Firewall sólo para que puedan hacer respaldos. En lugar de eso, haga que la máquina de Firewall confíe en el sistema interno ó mejor aún , ponga un lector de cintas local en la máquina para que pueda hacer sus propios respaldos.

Muchos de los problemas de seguridad comunes en Internet pueden considerarse fracasos por no seguir el principio de menor privilegio. Por ejemplo, existen y continuarán existiendo un sinnúmero de problemas de seguridad descubiertos en Sendmail un programa grande y complejo; con tales características es lógico que tenga problemas. El problema es que Sendmail ejecuta(por lo menos parte del tiempo) setuid como root; muchos de los ataques contra Sendmail aprovechan esto. Debido a que se ejecuta como root, Sendmail es un blanco de gran valor que llama mucho la atención de los atacantes; el hecho de que sea un programa cumplió sólo hace más fácil su trabajo. Esto implica que los programas privilegiados deben ser lo más sencillo posible y que, si un programa complejo requiere de privilegios, debe buscar formas de separar y aislar las partes que lo necesitan de las partes complejas .

Muchas de las soluciones que empleará para proteger su sitio son tácticas para reforzar con estrategia de menor privilegio. Por ejemplo, un sistema para filtrado de paquetes está diseñado para permitir la entrada de paquetes sólo para lo servicios que usted requiere. Ejecutar programas inseguros en un ambiente donde están disponibles sólo los privilegios que en realidad necesitan los programas (por poner un caso, una máquina que fue desbalijada d una manera u otra) es otro ejemplo; esta es la esencia de un host bastión.

Hay dos problemas al intentar cumplir con el menor privilegio. Primero, puede ser complejo implementarlo si aún no es una característica de los programas y los protocolos que usted utiliza. Si intenta agregarlo puede ser muy difícil que quede bien. Algunos automóviles que intenta poner en táctica el menor privilegio con llaves separadas para la cajuela y el encendido tienen botones para abrir de manera remota la cajuela que son accesibles sin las llaves o asiento traseros que se doblan y permiten el acceso a la cajuela sin abrirla en forma tradicional. Debe tener mucho cuidado para asegurarse de que en realidad ha tenido éxito al implantar el menor privilegio.

Segundo, tal vez acabe implementado algo menos que el menor privilegio. Algunos automóviles tienen un botón o una palanca para abrir el tapón de la gasolina en la guantera. Se supone que eso evita que los empleados del estacionamiento le roben la gasolina, pero si le presta aun amigo su auto, lo más probable es que quiera devolvérselo con el tanque de gasolina lleno. Si le da a su amigo la llave del encendido le está dando menos de privilegio mínimo que quiera que tenga (porque no podrá llenar el tanque de gasolina, pero si agrega la llave de la cajuela y la guantera puede más privilegio de los que quiere que tenga.

Puede encontrar efectos similares en las implementaciones de menor privilegio y las computadoras. Intentar cumplir con el menor privilegio con las personas, en lugar de programas, puede ser bastante peligroso. Puede predecir con facilidad los permisos que necesitará Sendmail para hacer su trabajo; los seres humanos son menos predecibles y más fácil que se molesten y se vuelvan peligrosos si no pueden hacer lo que quieren. Puede Tenga mucho cuidado para evitar que sus usuarios se conviertan en sus enemigos.

1.3.2 DEFENSA A FONDO

Otro principio de seguridad (de nuevo cualquier tipo de seguridad) es la defensa a fondo. No dependa solamente de un mecanismo de seguridad sin importan cuan fuerte parezca.; instale varios mecanismos que se respalden entre sí. No querrá que la falla de un solo mecanismo de seguridad comprometa por completo toda su seguridad. Puede ver aplicaciones de este principio en otros aspectos de la vida. Por ejemplo, es probable que su puerta tenga una chapa y un cerrojo con una barra; es probable que su automóvil tenga un seguro en la puerta y un seguro de encendido, etc.

Cualquier seguridad (hasta el Firewall que parezca más impenetrable) puede ser violado por atacantes dispuestos a tomar suficientes riegos y emplear suficiente fuerza. El truco está que en hacer el intento sea demasiado riesgoso o costoso para los atacantes que tal vez deban enfrentar. Puede hacer esto adoptando múltiples mecanismos que se den respaldo y redundancia entre sí: la seguridad de red (un Firewall), seguridad de host (en particular para su host bastión) y seguridad humana(educación del usuario, administración cuidadosa del sistema, etc). Todos estos mecanismos son importantes y pueden ser altamente eficaces, pero no ponga toda su fe sólo en uno de ellos.

Es probable que su Firewall en sí tenga varias capas. Por ejemplo, una arquitectura tiene varios filtros para paquetes; se configuran así porque se necesita dos filtros para hacer cosas diferentes, pero es bastante común configurar el segundo configurar el segundo para rechazar paquetes que el primero debió haber rechazado. Si el primer filtro está funcionando adecuadamente, esos paquetes jamás llegarán al segundo; sin embargo, si hay algún problema con el primero, todavía existe la esperanza de que esté protegido por el segundo.

He aquí otro ejemplo: si no quiere que las personas envíen correo a una máquina, no sólo utilice el filtro para los paquetes de salida, también quite los programas de correo de la máquina. En las situaciones donde el costo es reducido, siempre debe emplear defensas redundantes.

1.3.3 PUNTO DE CHOQUE

Un punto de choque obliga a los atacantes a utilizar un canal angosto que usted puede monitorear y controlar. Hay muchos ejemplos de choque en su vida: la caseta de un puente, las registradoras en un supermercado, la taquilla de un cine.

En la seguridad de redes, el Firewall entre su sitio e Internet (suponiendo que es la única conexión entre ambos) es el punto de choque; cualquiera que vaya a atacar su sitio desde Internet tendrá que pasar a través de ese canal, el cual debe estar defendido contra los ataques. Usted debe cuidarse de esos ataques y estar listo para responder si los detecta.

Un punto de choque es inservible si hay una manera efectiva de que un atacante lo evite. ¿Porqué molestarse en atacar la puerta principal, que está fortificada, si la puerta trasera esta totalmente abierta? De igual forma, desde el punto de vista de seguridad de red, ¿Porqué molestarse en atacar un Firewall si hay decenas o cientos de líneas conmutadas sin seguridad que pueden ser atacadas con mayor facilidad y tal vez con mayor éxito.

Una segunda conexión con Internet (e incluso indirecta); una conexión a otra compañía que tiene su propio enlace en otra parte; por poner un caso) es una violación a un más amenazante. Los atacantes que usan Internet como base quizá no dejen disponible un módem, o tal vez no hayan adquirido un servicio telefónico que no necesitan pagar, pero con toda certeza pueden encontrar conexiones redundantes de Internet hacia su sitio.

Un punto de choque quizá le aparezca como poner todos los huevos en una canasta y, por lo tanto, mala idea, pero la clave es que se trata de una canasta que puede proteger con sumo cuidado. La alternativa es dividir su atención entre muchos posibles frente de ataque. Si lo hace así; es probable que no pueda hacer un trabajo adecuado y no defienda bien los frentes de ataque, o que alguien pase por uno mientras este defendiendo otro (donde pueden incluso crear un falso ataque específicamente para distraer su atención del verdadero ataque).

1.3.4 ESLABÓN MÁS DÉBIL.

El punto fundamental de seguridad es que una cadena es tan fuerte como su eslabón más débil y una pared están fuerte como su punto más débil. Los atacantes inteligentes buscan el punto débil y concentran su atención en él. Debe reconocer los puntos débiles de su defensa y para que pueda monitorear con cuidado los que no pueda eliminar. Debe prestar igual atención a todos los aspectos de su seguridad para que no haya una gran diferencia en que tan inseguro es uno de ellos en comparación con otro.

Sin embargo, siempre hay un eslabón más débil, el truco consiste en hacer que sea lo suficientemente fuerte y mantenerlo así de acuerdo con el riesgo: Por ejemplo, es muy razonable preocuparse por personas que lo atacan a través de la red que por las que van a su sitio a atacarlo físicamente; por lo tanto, puede permitir que su seguridad física sea su eslabón más débil. No es razonable descuidar totalmente la seguridad física, pues todavía existe una amenaza ahí.

Tampoco es razonable, por ejemplo, proteger las conexiones Telnet con mucho cuidado, pero no proteger las conexiones FTP, debido a los riesgos similares que representan estos servicios.

Los modelos de seguridad para host sufren de una interacción bastante desagradable entre puntos de choque y eslabones débiles; no hay puntos de choque, lo cual significa que existe un gran número de eslabones y muchos pueden ser en realidad muy débiles.

1.3.5 POSTURA DE FALLA SEGURA.

Otro principio fundamental de la seguridad es que, en la medida de lo posible, los sistemas deben tener una falla segura, es decir, si van a fallar deben hacerlo de tal forma que nieguen el acceso a un atacante en lugar de dejarlo entrar. La falla también causar la negación del acceso a usuarios legítimos hasta que se hagan las reparaciones, pero por lo general es algo aceptable.

Las fallas seguras son otro principio de amplia aplicación en lugares familiares en nuestra vida diaria. Los dispositivos eléctricos están diseñados para apagarse- detenerse- cuando fallan de alguna forma. Los elevadores están diseñados para asir sus cables si no tienen energía eléctrica. Los seguros de las puertas eléctricas por lo general se abren cuando falla la energía, a fin de evitar que las persona en los edificios queden atrapadas.

La mayoría de las aplicaciones que analizamos son de falla segura, por ejemplo si se descompone un router para filtrado de paquetes, no deja pasar ningún paquete. Si un programa proxy falla no proporciona servicio. Por otro lado algunos sistemas para filtrado de paquetes basados en host están diseñados para permitir que los paquetes lleguen a una máquina que ejecuta una aplicación o filtrado de paquetes e in dependiente ejecuta aplicaciones que proporcionan servicios. La forma en que funcionan algunos de estos sistemas es que en caso e que falle la aplicación para filtrado de paquetes (o nunca se inicie el momento de arrancar), estos serán entregados a las aplicaciones que proporcionan servicios. Esto no es un diseño de falla segura y debe evitarse.

La aplicación más importante de este principio en la seguridad para redes reside en seleccionar la postura de su sitio respecto a la seguridad. Su postura es, en esencia, la actitud en general de su sitio en relación con su seguridad. ¿Se inclina por ser permisivo o restrictivo? ¿Tiene más inclinación a fallar en dirección de la seguridad (algunos llaman a esto paranoia) o de la libertad?

Hay dos posturas fundamentales que puede adoptar con respecto a decisiones y políticas de seguridad:

- Postura de negación preestablecida: especifique sólo lo que permite y prohiba todo lo demás.
- Postura de permiso preestablecido: especifique sólo lo que prohibe y permita todo lo demás.

Tal vez le parezca obvio cuál de estos es el enfoque "correcto"; desde el punto de vista de la seguridad, es la postura de negación preestablecida. Es probable que también sea obvio para los usuarios y administradores; desde su punto de vista, la postura correcta es la de permiso preestablecido.

Postura de negación preestablecida:

Lo que no está permitido expresamente está prohibido.

La postura de negación preestablecida tiene sentido desde el punto de vista de la seguridad porque es una postura de falla segura. Acepta que lo que usted no conoce puede dañarlo. Es la opción obvia más segura para la mayoría de las personas, pero en general no lo es para los usuarios.

Con la postura de negación preestablecida, usted prohibe todo por omisión; después, para determinar lo que va a permitir, usted debe:

- Examinar los servicios que necesitan sus usuarios.
- Considerar cómo afectarían la seguridad tales servicios y cómo puede proporcionarlos de manera segura.
- Permitir sólo los servicios que comprende, que puede proporcionar con seguridad y para los cuales ve una necesidad legítima.

Los servicios se activan basándose en cada caso. Empiece por analizar la seguridad de un servicio específico y haga un balance comparado los efectos que tendría en la seguridad contra las necesidades de los usuarios. Basándose en este análisis y en la disponibilidad de varios remedios para mejorar la seguridad del servicio, opte por un compromiso adecuado.

Para un servicio, podría determinar que éste se proporcione a todos los usuarios con toda seguridad mediante el filtrado de paquetes o con los sistemas proxy, disponibles fácilmente en el mercado. Para otro servicio, podría determinar que no es adecuadamente seguro por los medios disponibles en la actualidad, pero que un pequeño número de usuarios o sistemas lo requieren. En este último caso, quizá pueda restringir su uso a ese pequeño conjunto de usuarios (que pueden cobrar conciencia de los riesgos mediante entrenamiento especial) o sistemas (que puede proteger de otras maneras, por ejemplo, a través de seguridad especial para host). La clave es encontrar un punto intermedio que se adapte a su situación particular.

Postura de permiso preestablecido:

Lo que no está prohibido expresamente está permitido.

La mayoría de los usuarios y administradores prefieren la postura de permiso preestablecido. Tienden a suponer que todo estará, por omisión, permitido, y que se irán prohibiendo ciertas acciones y servicios problemáticos específicos conforme sea necesario. Por ejemplo:

- NFS no está permitido a través de un Firewall.
- El acceso a WWW está restringido a usuarios que han recibido capacitación sobre los problemas de seguridad que implica.
- Los usuarios no tienen permiso de instalar servidores no autorizados.

Ellos quieren que usted les diga qué es peligroso; que les enumere esas pocas (según ellos) pocas que no pueden hacer y que les deje hacer todo lo demás. Esto, en definitiva, no es una postura de falla segura.

Primero, supone que usted conoce de antemano y de manera precisa cuáles son los peligros específicos, como explicarlos para que los usuarios los comprendan y como protegerse contra ellos. Adivinar que peligros podrían estar en el sistema o en Internet es, en esencia, una tarea imposible. Sencillamente hay demasiados problemas posibles y demasiada información (por decir algo, nuevos agujeros en la seguridad, nuevas formas de explorar agujeros antiguos, etc.) para mantenerse actualizado. Si no sabe que algo es un problema, no estará en su lista de "prohibido". En ese caso, continuará siendo un problema hasta que se dé cuenta de ello y es probable que se percate porque alguien se aprovecha de él.

Segundo, la postura de permiso preestablecido tiende a degenerar en una "carrera armamentista que continua creciendo entre quien le da mantenimiento al Firewall y los usuarios. Quien da mantenimiento prepara defensas contra la acción e inacción del usuario; los usuarios inventan formas nuevas, fascinantes e inseguras de hacer las cosas; y proceso se repite una y otra vez. Quién da mantenimiento siempre intenta mantenerse al día. Es inevitable que existan periodos de vulnerabilidad entre el momento en que se instala un sistema, el momento en que se descubre un problema de seguridad y el momento en que la persona que le da mantenimiento puede responderle al problema. No importa cuan cuidadosos y cooperativos sean todos, algunas cosas se van a colar siempre en tales periodos: porque quién da mantenimiento nunca ha sabido de ellas, porque nunca se ha dado cuenta de todas las consecuencias para la seguridad, o porque sencillamente no ha tenido tiempo de resolver el problema.

Casi las únicas personas que se benefician de la postura de permiso preestablecido son los atacantes potenciales porque quién mantiene el Firewall no puede tapar todos los agujeros, siempre está en la modalidad de apagafuegos y quizá este demasiado ocupado para advertir las actividades del atacante.

Por ejemplo, considere el problema de compartir archivos con colaboradores en otros sitios. La primera idea del usuario será con certeza, utilizar la misma herramienta que emplean para compartir archivos internamente: NFS. El problema es que NFS es totalmente inseguro para ser permitido dentro a través de un Firewall. Supongamos que su postura es permisiva y no ha dicho específicamente a sus usuarios que no es seguro ejecutar NFS a través de un Firewall (o aunque se les haya dicho, no lo recuerdan o no les importa). En este caso, usted mismo podría encontrarse ejecutando NFS a través de su Firewall porque a alguien le pareció una buena idea, alguien que no comprendía (que no le importaron) los aspectos de la seguridad. Por otro lado, si su postura es de negación prestablecida, los intentos de sus usuarios por instalar NFS fallarán.

1.3.6 PARTICIPACIÓN UNIVERSAL.

Para que sean totalmente efectivos, la mayoría de los sistemas de seguridad requieren de participación universal (o por lo menos la ausencia de oposición activa) por parte del personal de un sitio. Si alguien opta simplemente por salirse de sus mecanismos de seguridad, entonces el atacante puede agredirlo a usted atacando primero el sistema exento de esa persona y luego su sitio desde adentro. Por ejemplo, el mejor Firewall del mundo no lo protegerá si alguien que lo ve como una carga excesiva instala una conexión trasera desde su sitio e Internet a fin de evitar el Firewall. Esto puede ser tan fácil como comprar un módem obtener gratis un software PPP o

SLIP de Internet y pagar unos dólares al mes a un proveedor local de servicios de Internet de baja velocidad; esto está dentro del alcance de precios y habilidades técnicas de muchos usuarios y administradores.

Incluso formas más mundanas de rebelión pueden arruinar su seguridad. Necesita que todos notifiquen ocurrencias extrañas que pueden estar relacionada en la seguridad; usted no puede ver todo. Necesita que las personas seleccionen buenas contraseñas; que las cambien con regularidad , que no se las den a amigos, pariente o mascotas.

¿Cómo hacer para que todos participen? La participación puede ser voluntaria (convenza a todos de que es una buena idea), involuntaria (alguien con suficiente autoridad y poder les dicen que tienen que cooperar o de lo contrario ...), o una combinación de ambas. Es obvio que la participación voluntaria es preferible a la involuntaria; querrá que la gente le ayude; no que busque formas de evitarlo. Esto significa que quizá deba trabajar intensamente dentro de su organización convenciendo a las personas de las ventajas de la seguridad y de que los beneficios sobrepasan los costos.

Las personas que no son participantes voluntarios se tomarán muchas molestias para evitar las medidas de seguridad. En un sistema de correo de voz que requería que se cambiaran las contraseñas cada mes, varias personas descubrieron que registraba sólo seis contraseñas viejas, y les dio por cambiarlas siete veces seguidas (En siete llamadas telefónicas independientes!) para poder utilizar la misma contraseña. Este tipo de comportamiento conduce a una carrera armamentista (los programadores limitan el número de veces que pueda cambiar su contraseña) y pronto varias persona se ven envueltas en una batalla totalmente interna. Usted tiene mejores cosas que hacer con su tiempo, así como sus usuarios, vale la pena gastar mucha energía para convencer a las personas de que cooperen voluntariamente, pues con frecuencia necesitará igual energía para obligarlos, con peores efectos secundarios.

1.3.7 DIVERSIFICACIÓN DE DEFENSA.

Así como puede tener seguridad adicional utilizando varios sistemas para dar profundidad a su defensa, también puede obtenerla empleando varios tipos de sistemas. Todos sus sistemas son iguales, alguien que pueda entrar a alguno de ellos quizá penetre a todos.

La idea que sirve de apoyo a la diversificación de defensa es que utilizar sistemas de seguridad de diferentes proveedores puede reducir las posibilidades de un problema o error de configuración común que pueda comprometerlos a todos. Sin embargo, hay un balance en términos de complejidad y costo. Procurar e instalar varios sistemas diferentes es más difícil, toma más tiempo y es más costoso que procurar e instalar un solo sistema (o incluso varios sistemas idénticos). Tendrá que comprar los múltiples sistemas (con menor descuento de cada proveedor porque le está comprando menos) y múltiples contratos de mantenimiento para protegerlos. También le tomará tiempo y esfuerzo adicional para que su equipo humano aprenda a manejar estos sistemas diferentes.

Tenga cuidado con la diversidad ilusoria. Sólo por utilizar sistemas UNIX de diferentes proveedores no compra la diversidad, pues la mayoría de ellos deriva del código fuente de BSD o System V. Además, las aplicaciones para redes UNIX más comunes (Sendmail, Telnet/Telned, FTP/FTPD, etc.) derivan de las fuentes de BSD, sin importar si está en una plataforma basada en BSD o SystemV. Hubo un sinnúmero de errores y problemas de seguridad en los elementos

originales, los cuales se propagaron en la mayoría de las versiones específicas de cada proveedor de estos sistemas operativos; muchas versiones de UNIX específicas de un proveedor aún tienen errores y problemas de seguridad que se descubrieron por primar vez hace algunos años en otras versiones de otros proveedores y aún no se han arreglado.

También debe saber que varios sistemas configurados por la misma persona (o por el mismo grupo de personas) pueden compartir problemas comunes si estos radican en bases tecnológicas no conceptuales. Si el problema es un malentendido sobre como funciona un protocolo específico todos sus sistemas pueden estar configurados incorrectamente, ya que se configuraron de la misma forma, siguiendo ese malentendido.

1.4 SERVICIOS COMUNES DE INTERNET

Hay un sin número de servicios estándar de Internet que los usuarios utilizan y que la mayoría de los sitios intentan soportar. Existen razones importantes para utilizar tales servicios; de hecho, sin ellos hay pocas razones para conectarse a Internet. Pero también existen problemas de seguridad potenciales con cada uno de ellos.

¿Qué servicios quiere soportar en su sitio? ¿Cuáles puede soportar de manera segura? Cada sitio es diferente. Cada uno tiene su propia política de seguridad y su propio ambiente de trabajo. Por ejemplo, ¿todos usuarios necesitan correo electrónico? ¿Todos necesitan transferir archivos fuera de su organización? ¿Qué tal descargar archivos desde sitios fuera de la propia red de la organización? ¿Quién puede iniciar una sesión remota desde otra ubicación usando Internet?

Ninguno de estos servicios es, en realidad, seguro; cada uno tiene sus propias debilidades y cada uno ha sido explotado de varias formas por sus propios atacantes. Antes de que decida soportar un servicio en su sitio, debe evaluar que tan importante para sus usuarios y si podrá protegerlos de sus peligros. Hay varias formas de hacerlo: ejecutar los servicios en ciertas máquinas protegidas, empleando variaciones especialmente seguras de los servicios estándar; ó, en algunos casos, bloquear los servicios por completo desde o hacia algunos o todos los sistemas externos.

1.4.1 CORREO ELECTRÓNICO

El correo electrónico es uno de los servicios de redes más populares y básicos. Es de riesgo relativamente bajo, pero eso no significa que esté libre de riesgos. Falsificar correo electrónico es sencillo (como lo es falsificar el correo postal normal), y las falsificaciones facilitan dos tipos de ataques: ataques contra su reputación y ataques de manipulación social (por poner un caso, ataques en los que los usuarios envían correo que se supone viene de un administrador, aconsejándoles a otros que cambien su contraseña de forma específica). Aceptar correo electrónico ocupa tiempo en la computadora y espacio en el disco exponiéndolo a ataque s de negación del servicio; con una configuración adecuada, sólo se negará el servicio de correo electrónico. En particular con sistemas modernos de correo multimedia, las personas pueden enviar correo electrónico que contenga problemas que si se ejecutan con supervisión insuficiente pueden resultar caballos de Troya.

Aunque la gente se preocupa más sobre el último riesgo mencionado, en la práctica los problemas comunes con el correo electrónico son inundaciones inadvertidas (incluyendo cadenas de cartas) y personas que confían plenamente en la confidencialidad del sistema de correo

electrónico y envían sus datos por medio del correo de Internet. Sin embargo, aunque los usuarios sean educados el servicio de correo se aísle de los demás servicios para que los atacantes de negación de servicio inadvertidos a propósito afecten lo menos posible, el correo es razonablemente seguro.

El protocolo simple de transferencia de correo (SMTP) es el protocolo estándar de Internet para enviar y recibir correo electrónico. SMTP en si 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 para los atacantes.

El servidor SMTP más común en UNIX es Sendmail. Sendmail se ha explotado con un gran número de casos de accesos ilegales (incluyendo el gusano de Internet), por lo que las personas se ponen nerviosas al utilizarlo. Sin embargo, muchos de sus posibles sustitutos no gozan de gran preferencia; la evidencia sugiere que están menos explotados porque son menos populares no porque sean menos vulnerables. Hay excepciones en los programas diseñados explícitamente para seguridad, pero no soportan todas las funciones necesarias para enviar y recibir mensajes de correo arbitrarios; algunas cosas se manejan mejor con Sendmail ejecutado con un espacio seguro.

1.4.2 TRANSFERENCIA DE ARCHIVOS

El correo electrónico transfiere datos de un lugar a otro pero está diseñado para archivos pequeños legibles para las personas. Los protocolos para la transferencia de correo electrónico tienen permitido hacer cambios a un mensaje que son aceptables para las personas (por ejemplo, insertar el signo ">" antes de la palabra " from (de) al principio de una línea para quien envíe el mensaje no se confunda con una línea de encabezados, pero que no los son para los programas.

Aunque los sistemas de correo electrónico actuales incluyen algoritmos elaborados para tales problemas, de tal forma que en un archivo binario puede dividirse en piezas pequeñas y codificarse en el extremo que envía y decodificarse y reensamblarse en el que recibe estos algoritmos son engorrosos y propensos a errores. Además, las personas quizás quieran salir y buscar los archivos de manera activa, en lugar de esperar a que alguien los envíe. Por lo tanto, aun cuando el correo electrónico está disponible es útil tener un método diseñado para transferir archivos al solicitarlos.

El protocolo de transferencia de archivos (FTP) es el protocolo estándar de Internet para este propósito. En teoría , permitir que sus usuarios obtengan archivos no incrementa más el riesgo que permitir el correo electrónico de hecho, algunos sitios ofrecen servicios que permiten que tenga acceso a FTP por medio de correo electrónico. En la práctica, sin embargo, las personas realizaran más transferencias de archivos cuando esté disponible, y más probable que obtengan programas y datos indeseables.

¿Qué hace que estos programas y estos datos sean indeseables? La preocupación principal en la mayoría de los sitios es que los usuarios obtengan software tipo caballos de Troya. Aunque esto puede suceder en la realidad la mayor preocupación es que los usuarios obtengan juegos de computadora, software pirata e imágenes pornográficas, que tienden a ocupar una cantidad molesta de tiempo y espacio en el disco, pero no representan un riesgo importante de seguridad.

Si se asegura de hacer lo siguiente entonces puede considerar a FTP un servicio razonablemente seguro que facilita a recursos importantes en Internet:

- Eduque a sus usuarios para que desconfíen de cualquier software que obtengan por medio de FTP.

- Comunique a sus usuarios las políticas de sus sitio sobre acoso sexual y el uso de recursos de su organización.

¿Qué pasa con el otro lado de la moneda: permitir a otras personas utilizar FTP para transferir archivos desde sus computadoras? Esto es más riesgoso el FTP anónimo (anonymous ftp) es un mecanismo extremadamente popular de dar acceso a usuarios remotos a los archivos sin tener que darles un acceso completo a su máquina. Si ejecuta un servidor FTP, puede permitir que los usuarios obtengan archivos colocados en un área colocada separada de su sistema sin dejarlos iniciar una sesión, y potencialmente, tener acceso a todo sus sistema. El área de FTP anónimo de sus sitio puede ser el archivo público de documentos estándares, software, imágenes e información de otros tipos dentro de sus organización que las personas necesitan de usted o que usted quiere compartir con ellos. Para muchas organizaciones el establecimiento de un sitio FTP es el primer paso apara realzar negocios por Internet.

Para tener acceso a los archivos que ha de tener disponibles, los usuarios inician una sesión en sus sistema utilizando la clave especial para el acceso FTP: "anonymous". La mayor parte de los sitios piden que los usuarios tecleen su dirección de correo electrónico, como respuesta a la solicitud de contraseña como una cortesía para que el sitio pueda dar seguimiento a quien utiliza el servicio FTP anónimo, pero este requisito rara vez se hace cumplir (en gran parte porque no hay una manera fácil de verificar la validez de una dirección de correo).

Para instalar un servidor FTP anónimo, debe asegurarse que las personas que lo utilicen no pueden tener acceso a otras áreas o archivos del sistema y que no puedan utilizar FTP para el acceso al sistema en si. Los directorios que pueden escribirse en el área de FTP anónimo son una preocupación especial.

También tendrá que asegurarse de que sus usuarios no utilicen el servicio de manera apropiada.

Puede ser muy tentador para las personas poner archivos que quieren que lean personas específicas.

Muchas veces las personas no recapacitan en que cualquiera en Internet pueden leerlos o si lo hacen, pero creen en la seguridad de ser desconocido. Por desgracia para estos inocentes un sinnúmero de herramientas intentan indizar servidores FTP anónimo y tienen éxito en deshacerse de casi todo lo desconocido.

Quizá haya escuchado de otros protocolos para la transferencia de archivos el Protocolo Trivial de Archivos (TFTP) es un protocolo FTP simplificado que las máquinas sin disco utilizan para transferir información. Es en extremo sencillo integrarlo al hardware y , por lo tanto, no soporta ninguna autenticación. No hay razón para proporcionar acceso TFTOP fuera de su red; los usuarios comunes no transfieren archivos con ese protocolo.

UUCP (UNIX CoPy) es un protocolo más antiguos para transferir archivos a través de módems, y a veces todavía se utiliza para transferir noticias de USENET y correo electrónico, aún a través de Internet, en particular para los sitios con conexiones intermitentes a esa red. Tales sitios utilizan UUCP a través (montando en) TCP para obtener correo electrónico de proveedor de servicios cada vez que se conectan a la red. Aunque antes era común pocos sitios proporcionan servicios UCCP anónimo para el acceso a archivos. Como protocolo para transferencia de archivos, es de interés para quienes no tienen una conexión a Internet y utilizan módems. A no ser que todo su negocio pretenda proporcionar información pública, UUCP probablemente no es útil para sus usuarios exteŕnos.

El protocolo de servicios de archivos (FSP) es u protocolo para transferir archivos desarrollado para circundar las restricciones de FTP. Los usuarios no comunes no pueden instalar servidores FTP, pero si fsp y las transferencias fsp pasan cuando las transferencias FTP han sido bloqueadas. Hay dos desventajas para instalar un servidor fsp:

La seguridad no es una preocupación específica para diseñadores y usuarios de fsp; aunque debería reducir inherentemente la seguridad de su máquina es susceptible a los diferentes de problemas de FTP, y se han hecho menos intentos para evitarlos o corregirlos.

Proporcionar el servicio fsp no le va a hacer amigos entre los administradores de sistemas remotos, ya que el objetivo de fsp es permitir que las personas transfieran archivos después de que sus sitios han decidido bloquear la transferencias de archivos.

Fsp si tiene algunas ventajas reales sobre FTP (diseñado para obrar ocultamente lo convierte en un "usador" de recursos poco frecuente), pero casi nadie lo use y su historia hace poco probable que en el futuro haya muchos usuarios.

Dentro de un sitio, quizá desee utilizar rcp para transferir archivos entre sistemas. Rcp (descrito con el resto de los así llamados "comandos r de Berkeley") es un programa para transferir archivos que se comporta como una versión extendida del comando C.P. de UNIX. Es inapropiado utilizarlo a través de Internet porque emplea un modelo de autenticación basado en el anfitrión. En lugar de requerir autenticación del usuario en la máquina remota, ve la dirección IP del anfitrión de quien viene la solicitud. Por desgracia, no puede saber cuáles paquetes vienen en realidad de ese anfitrión.

1.4.3 ACCESO DE TERMINAL REMOTA Y EJECUCIÓN DE COMANDOS.

Los programas que proporcionan acceso de terminal remota permiten que utilicen un sistema remoto como si su máquina fuera una terminal conectada directamente.

Telnet es el estándar para el acceso de terminal remota en Internet. En verdad imita una terminal, no una estación de trabajo gráfica; proporciona acceso a aplicaciones basadas en caracteres. También brinda acceso remoto a sus usuarios desde cualquier sitio conectado a Internet sin hacer arreglos especiales.

Telnet se considero en un tiempo un servicio más o menos seguro porque requiere que los usuarios se autentiquen por ellos mismos. Por desgracia, Telnet envía toda su información sin codificar, lo que lo hace muy vulnerable a ataques de espionaje (utilizando analizadores de protocolo) y robo. Por esta razón, ahora Telnet se considera de los servicios más peligrosos cuando se utiliza para entrar a un sitio desde sistemas remotos (entrar a sistemas remotos desde su sitio es problemas de ellos no suyo). Telnet es seguro sólo si la máquina remota y todas las redes entre ella la máquina local son seguras lo cual significa que no es seguro a través de Internet, donde no pueden identificar con certeza las redes que intervienen, mucho menos confiar en ellas. Por otro lado, Telnet puede ser muy útil (y en extremo efectivo en cuanto costos) como un mecanismo de acceso remoto si sus usuarios viajan con frecuencia a sitios conectados a Internet. En los lugares donde utilizar un módem es costoso, difícil y lento, usar una conexión a Internet por medio de Telnet puede ser la mejor solución, por esta razón práctica tal vez deba proporcionar el servicio de Telnet, pero con precaución.

Hay varios tipos de esquemas para dar autenticación a inicios de sesión remotos, pero aunque la autenticación protege su contraseña aún así su sesión puede ser intervenida o robada.

Existen programas, además de Telnet que pueden usarse para tener acceso como terminal remota y ejecución remota de programas (rlogin, rsh, on). Estos programas se utilizan en un ambiente confiable para permitir que los sus usuarios tengan acceso remoto sin que deban autenticarse nuevamente. El anfitrión al que se conectan confía en que el sistema solicitante ha dado autenticación al usuario en forma correcta. El modelo de la autenticación basado en el anfitrión es simplemente inapropiado para utilizarse a través de Internet porque, en general, no puede confiar en los hosts fuera de su red. De hecho, ni siquiera puede estar seguro de que los paquetes sean del host que dicen ser.

Rlogin y rsh pueden ser apropiados para utilizarse dentro de una red protegida por un Firewall, dependiendo de sus políticas internas de seguridad. Sin embargo, no coloca todas sus inspecciones de seguridad en el programa cliente, y cualquiera puede utilizar un cliente modificado que salte estas inspecciones, por lo que es totalmente inseguro para emplearse aún dentro de una red de área local protegida por un Firewall (permite que cualquier usuario ejecute un comando como otro usuario). Usted puede desactivar la aplicación on desactivando el servidor red.

1.4.4 NOTICIAS DE USENET

Mientras el correo electrónico permite a las personas comunicarse, es más eficaz para que una persona envíe su mensaje a otra, o una pequeña lista de personas interesadas en un tema específico. Los grupos de noticias (newsgroups) son la contraparte en Internet de los tableros de foros de discusión (bulletin boards), y están diseñados para comunicación de muchos a muchos. Las listas de distribución de correo también soportan comunicación de muchos a muchos, pero de manera menos abierta y eficaz, ya que no hay forma fácil de saber sobre todas las lista de distribución de correo, y cada receptor tiene su propia forma de cada mensaje. Las listas de correo de discusión más grandes (por ejemplo, listas donde se realizan discusiones entre los suscriptores, en lugar de las utilizadas sólo para distribuir información o anuncios a los suscriptores) tiene miles de suscriptores; los grupos de noticias más populares tienen cientos de miles. Las noticias de Usenet son como la televisión: suceden muchas cosas; la mayor parte tiene poco valor social; una parte es fantásticamente divertida o informativa, y todos la quieren.

Los riesgos de las noticias son muy similares a los del correo electrónico: sus usuarios pueden, tontamente, confiar en la información recibida; pueden divulgar información confidencial y usted puede inundarse de mensajes. Las noticias asemejan una inundación cuando funcionan de forma normal (la mayoría de los sitios reciben ciento de megabytes al día, y la cantidad se incrementa en forma constante, duplicando su volumen aproximadamente cada seis meses), así que debe asegurarse de configurarlas pare que las inundaciones no afecten otros servicios. Debido a que las noticias rara vez son un servicio socia, los ataques de negación del servicio en un solo sitio en general se ignoran. Los riesgos de la seguridad de las noticias son, por lo tanto bastante bajos. Quizá desee evitar las noticias porque no tiene el ancho de banda o el espacio en disco necesario, pero no es un problema de seguridad importante.

El protocolo de transferencia de noticias en red (NNTP?) se utiliza para transferir noticias a través de Internet. Para instalar el servidor de noticias en su sitio debe de determinar la forma más segura de que fluyan las noticias a sus sistemas internos para que NNTP no pueda ser utilizado para penetrar su sistema principal. Algunos sitios ponen el servidor de noticias en el host bastión (el más confiable), otros en un sistema interno NNTP no hace mucho, y sus transferencias externas de noticias son con otra máquinas especificas (no es como el correo, del que se espera recibir información de todos), así que no es muy difícil de asegurar.

El elemento de seguridad más importante al que se enfrenta con las noticias es que hacer con grupos de noticias privados. Muchos sitios crean grupos locales privados para facilitar las discusiones entre sus usuarios; estos grupos con frecuencia contienen información delicada, confidencial o propietaria. Alguien con acceso a sus servidores NNTP, puede, en potencia, acceder a estos grupos de noticias privados, lo que viene en divulgación de esta información. Si va a crear un grupo de noticias privado, asegúrese de configurar NNTRP con cuidado para controlar el acceso a estos grupos.

1.4.5 WORLD WIDE WEB

El correo, FTP Telnet y las noticias de Usenet han existido desde los primeros días de Internet; en realidad, son extensiones de servicios proporcionados mucho antes de que existiera esa red. El (WWW) es un concepto nuevo basado totalmente en Internet y, en parte, en servicios existentes y en un protocolo nuevo: el protocolo de transferencia de hipertexto (HTTP).

Muchas personas confunden las funciones y el origen del WWW, Mosaic y HTTP, y la terminología que usan para referirse para estos tres elementos se ha vuelto confusa. Parte de la confusión se introdujo con intención; los programas de navegación para el web (llamados navegadores) intentan proporcionar una interface transparente para una gran variedad de información a través de una amplia variedad de mecanismos, y confundir las distinciones hacc su uso más fácil, aunque sea más difícil de comprender).

El WWW es la colección de servidores de HTTP en Internet. El web es responsable, en gran medida de la reciente explosión de actividad dentro de Internet. Se basa en conceptos desarrollados en el European Particle Physics Laboratory (CERN) en Ginebra, Suiza, por Tim Berners-lee y otros. La mayor parte del trabajo con los clientes web se llevó a cabo en el National Center for Supercomputing Apliccation (NCSA) en la universidad de Illinois, en Urbana-Cahampaing.

Hay muchas organizaciones e individuos desarrollando software para clientes y servidores web en estos días, y muchos más que usan estas tecnologías para una gran variedad de propósitos. Nadie "controla" el web así como nadie controla Internet.

El web utiliza tecnología de hipertexto para enlazar una gran cantidad de documentos que pueden incluir texto, imágenes, sonido, vídeo y otros formatos. Puede "navegar" por los documentos de cualquier manera (no sólo jerárquicamente) para buscar información el hipertexto proporciona la posibilidad de ir (navegar) de un documento a otro en Internet. Los usuarios pueden moverse libremente de uno a otro, sin importar en donde están guardados, con sólo hacer clic en una palabra o imagen para cual ha sido definido un enlace (o liga HTTP).

HTTP es el principal protocolo de aplicación que utiliza el World Wide Web: proporciona acceso de usuarios a los archivos que conforman el servicio web. Como ya se mencionó estos archivos pueden estar en múltiples formatos (texto, imágenes, audio, vídeo etc.), pero el más común en el servicio web es el lenguaje para marcar hipertexto (HTML). HTML es un lenguaje estándar para crear páginas web.

Proporciona capacidades básicas para dar formato a documentos (incluyendo la habilidad de incluir gráficas) y permite especificar enlaces de hipertexto a otros servidores y archivos.

Los navegadores web son increíblemente populares, y con razón. Proporcionan una interface gráfica rica en un gran número de servicios de Internet. La información y los servicios que no están disponibles antes o que eran sólo para expertos ahora son fácilmente accesibles. En Silicon Valley puede usar el web para que le entreguen la cena sin dejar la computadora excepto para abrir la puerta. Es difícil de percibir el web sin experimentarlo; abarca toda la gama de lo que puede hacer con una computadora desde lo mundano a lo sublime con un viaje extra a lo ridículo.

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 esta dificulta su control. Así como es más fácil transferir y ejecutar el programa correcto utilizando un navegador web que por medio de FTP, es más fácil transferir y ejecutar un o peligroso. Los navegadores dependen de programas externo, llamados genéricamente visualizadores ("viewers"), aunque reproduzcan sonidos en lugar de mostrar imágenes, para manejar los tipos de información que no entienden pos í solos. Los navegadores por lo común comprenden tipos de datos básicos, como html, texto sencillo e imágenes jpeg y gif. Debe tener mucho cuidado sobre que visualizador configura de modo predeterminado; no le agradaría un visualizador que pudiera hacer cosas peligrosas porque está ejecutándose en sus computadoras como si fuera uno de sus usuarios, ejecutando comandos de una fuente externa. También querrá advertir a los usuarios que no agreguen visualizadores o cambien las configuraciones de los mismos basándose en los consejos de extraños.

Debido a que un documento de HTML puede enlazarce con facilidad a documentos con otros servidores, es muy fácil que las personas se confundan sobre quien es exactamente responsable de un documento específico. Los usuarios nuevos quizá no se den cuenta cuando van de documentos internos en su sitio a documentos externos. Esto tiene dos consecuencias desafortunadas. Primero puede confiar en los documentos externos indebidamente (porque piensan que son documentos internos). Segundo, tal vez culpen a quienes administren el web interno por los pecados del mundo. A las personas que entienden el web les es difícil creer esto, pero es una equivocación, el lado obscuro de tener una transición suave entre sitios.

La mayoría de los servidores web son razonablemente seguros tal como se encuentran. Sin embargo, también pueden llamar programas externos, los cuales son más o menos fáciles de escribir pero muy difíciles de asegurar. Debe tratar cualquier extensión con la misma precaución con la que trataría a un servidor nuevo de cualquier tipo.

1.4.6 OTROS SERVICIOS DE INFORMACIÓN.

Muchos usuarios quieren tener acceso a servicios adicionales de información; Gopher, Wais y Archie son los más populares.

Gopher una herramienta orientada a menús, basada en texto, que ayuda a los usuarios a encontrar información en Internet. "Gopher no es acrónimo de nada; se desarrolló en la universidad de Minnesota cuya mascota es el Gopher dorado (castor dorado),. La información de un servidos 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 elemento. Hay varios clientes Gopher disponibles incluyendo shareware y clientes comerciales para Windows Mcintosh y UNIX. Los clientes y servidores Gopher emplea una serie de datos extensible, muy parecido a lo que hacen los clientes y servidores web, y por lo tanto, están sujetos a muchas de las mismas preocupaciones de seguridad.

El servicio de información de área amplia (WAIS) fue desarrollado por un consorcio de compañías: Thinking-Machines, Apple, Dow Jones y KPMG Peat Marwick; Brewster Kahle dirigió su desarrollo. Un usuario WAIS 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 el marcador. Este marcador se construye con el numero de veces que las palabras se mencionan y el tamaño del documento; los documentos más breves obtienen marcadores más altos al igual que los documentos que mencionan con más frecuencia las palabras claves.. La lista de documentos de vuelta esta clasificada por su marcador, para que con algo d suerte los documentos más relevante aparezcan primero. El servidor mantiene índices extensivos del contenido de todos los documentos que están en el servidor para permitirle hacer estas búsquedas de manera eficiente. En la actualidad hay cientos de servidores WAIS en Internet. Puede tener acceso a ellos con clientes WAIS o usando navegadores para entrar a sitios que proporcionan compuertas http-wais (por poner un caso, http://www.ai.mit.edu/the-net/wais.html).

Archie es un servicio de Internet que busca en los índices de servidores ftp anónimo los nombres de archivos y directorios. Es usual que los servidores Archie proporcionen el servicio a través de telnet y correo electrónico, además de los clientes Archie. Los proveedores de servicio Archie por lo general prefieren que los usuarios utilicen clientes Archie porque imponen menos carga en el servidor. Archie también es accesible por medio de navegadores web a través de sitios que proporcionan compuertas http-Archie, como la que esta http://www.nexsor.com.uk/archie.ahlhtml. En est momento sólo hay alrededor de 20 servicios Archie en todo el mundo en parte por los significativos recursos (poder de cómputo, espacio en disco, ancho de banda de red y tiempo del administrador) necesarios para ejecutar un servidor Archie, y en parte porque cada servidor Archie busca en casi en toda Internet archivos accesibles vía FTP. Si hubiera una gran cantidad de servidores Archie se consumiría tanto ancho de banda buscando información que sería imposible utilizar los recursos.

Wais y Archie están menos abiertos a travesuras que HTTP y Gopher ya que no devuelven datos de tipo arbitrario. Si un servidor Wais aconseja que un documento es sobre jardinería y resulta que es sobre fabricación de joyería en el tiempo libre, puede ser molesto, pero no es un reto a la seguridad de sus computadoras. Por desgracia, proporcionar acceso a uno de estos servicios puede abrir otros agujeros de seguridad no relacionados con el servicio en si,. Por ejemplo, permitir que sus usuarios tengan acceso a Archie directamente, permite que los atacantes tengan acceso a sus servidores NFS y NIS/YP.

Ejecutar servidores es algo más riesgoso. A diferencia de los clientes, los servidores para estos protocolos (incluyendo WAIS y Archie) aceptan consultas arbitrarias, y debe asegurarse de que no van a producir resultados inesperados. Cualquier servidor que actúa sobre solicitudes de usuarios potencialmente hostiles es vulnerables a ataques de negación del servicio y a ejecución de comandos inesperados en la máquina del servidor con permiso del programa servidor.

1.4.7 SERVICIOS DE NOMBRES

El servicio de nombres se encarga de traducir los nombres de host que utilizan las personas a las direcciones IP numéricas que utilizan las máquinas. En los primeros días de Internet, era posible para cada sitio mantener una tabla de hosts con el nombre y número de cada máquina en Internet que algunas vez podían encontrar interesante. Con millones de hosts enlazados no resulta práctico para ningún sitio mantener una lista semejante, mucho menos que la tenga cada sitio. En lugar de eso, el servicio de nombres de dominios (DNS) permite que cada sitio tenga información sobre sus propios hosts que puedan encontrar la información para otros sitios. DNS no es un servicio a nivel usuario en sí mismo, pero soporta SMTP, FTP, Telnet y casi cualquier otro sitio que necesiten los usuarios, quienes quieren escribir "Telnet fictional.com" en lugar de Telnet 10.100.242.32. Además muchos servidores de FTP anónimo no permiten conexiones de clientes a no ser que puedan utilizar un DNS para buscar el nombre de host del cliente a fin de iniciar la sesión.

El resultado neto es que debe utilizar y proporcionar el servicio de nombres para poder participar en Internet. El riesgo principal de proporcionar el servicio DNS es que de mas información de la pensada. Por ejemplo, DNS le permite incluir información sobre que hardware y software esta ejecutando lo cual no conviene que sepa el atacante. De hecho, tal vez ni siquiera desee que atacante conozca los nombres de todas sus máquinas internas.

Usar DNS interno y luego depender los nombres de host para dar autenticación lo hace vulnerable a un intruso que puede instalar un servicio de DNS mentiroso. Esto se puede manejar combinando algunos métodos, incluyendo:

1. Usar direcciones IP (en lugar de nombres de host) para dar autenticación a los servicios que deben ser más seguros.

2. Dar autenticación a usuarios en lugar de hosts en los servicios más seguros, porque las direcciones IP también pueden falsificarse.

Algunos sitios utilizan el servicio de información de red de Sun, antes conocidos como Yellow Pages (NIS/YP) para distribuir información de nombres de hosts internamente. No es necesario hacer esto: puede utilizar clientes DNS en lugar de cualquier plataforma que soporte NIS/YP, pero puede ser más conveniente para configurar su máquinas internas. Es cierto que no es necesario ni aconsejable proporcionar servicios NIS/YP a máquinas externas. Pues está diseñado para administrar u sólo sitio, no intercambiar información entre sitios, y es altamente inseguro. Por ejemplo, no sería posible proporcionan información de su host a sitios externos por medio de NIS/YP sin también proporcionar sus archivo de contraseña si ambos están disponibles internamente.

1.4.8 SERVICIOS PARA ADMINISTRACIÓN DE REDES.

Existe una variedad de ser vicios utilizados para administrar y mantener las redes; son servicios que la mayoría de los usuarios no emplean de manera directa (de hecho, muchos de ellos ni siquiera lo han oído mencionar) pero son herramientas muy importantes para los administradores de red.

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

Ping le dice si puede o no hacer llegar un paquete a y de un host determinado y, con frecuencia, información adicional, como cuanto tarda en hacer el viaje de ida y vuelta. Traceroute le notifica no sólo si puede llegar a un host específico (y se puede responder) sino además la ruta que siguen sus paquetes para llegar a {el, lo cual es muy útil para analizar problemas de la red en alguna parte entre usted y un destino.

Debido a que hay servidores para ping y traceroute, no es fácil decidir no encender los servidores. Es posible utilizar el filtrado de paquetes para evitar que estos se transmitan desde su sitio o se reciban en él, pero en general no es necesario. No hay riesgos conocidos para ping o traceroute de salida, y muy pocos para los ping y traceroute que entran. Pueden utilizarze para ataques y negación del servicio, pero no más que otros protocolos. Más amenazante, pueden emplearse para determinar que hosts existen en su sitio como paso preliminar para atacarlos. Por esta razón, muchos sitios evitan o limitan los paquetes relevantes que van a entrar.

El protocolo simple de administración de redes (SNMP) es un protocolo diseñado para administrar la administración central de equipo de red (routers, gateways, ,hubs, computadoras centrales y hasta cierto punto hosts). Las estaciones de administración SNMP pueden solicitar información (si una interface esta arriba o abajo, cuantos bytes se han transferido a través de esa interface, cuantos errores ha habido en ella) del equipo de red por medio de SNMP dichas estaciones también pueden controlar ciertas funciones del equipo de red (cambiando una interface hacia arriba o hacia abajo, configurando sus parámetros, etc.) . El equipo de red también puede reportar información urgente (por ejemplo, que una línea esté descompuesta o que haya un número importante de errores en una línea específica a tales estaciones por medio de SNMP.

El riesgo principal de seguridad con SNMP es que alguna otra persona pueda asumir el equipo de red y reconfigurarlo para sus propios propósitos (desactivar el filtrado de paquetes, cambiar el enrutamiento, ó, simplemente destruir su configuración).

1.4.9 SERVICIOS DE HORA

El protocolo de hora de red /(NTP) es un servicio de Internet que pone a tiempo los relojes de un sistema con gran precisión. Sincronizar la hora entre diferentes máquinas es importante por muchas razones. Desde el punto de vista de la seguridad examinar las horas precisas anotadas en los archivos de registro de diferentes máquinas puede ayudarlo a analizar patrones de entradas indeseadas. Tener relojes sincronizados también es un requisito para evitar que los atacantes graben una interacción y luego lo repitan,; si los registros de horas están codificados en la interacción, serán correctos la segunda vez que la transacción se repita. Desde un punto de vista práctico, los relojes sincronizados también se requieren para utilizar NFS con éxito.

No tiene que utilizar NTP a través de Internet; sincronizará los relojes entre sí dentro de su sitio si es todo lo que quiere. La razón por la cual las personas utilizan Internet NTP desde Internet es que cierto número de hosts en con relojes en extremos preciso (relojes de radio que reciben la señal de la hora de los relojes atómicos maestro de los Estados Unidos ó de los relojes atómicos que están en los satélites GPS, proporcionan servicio NTP para asegurarse de que sus relojes no sólo estén sincronizados entre sí sino también estén correctos. Si un servicio de hora externo, hora externa, quizá le ocurra que todas sus computadoras tienden exactamente la misma hora equivocada. Aceptar un servicio externo lo hace vulnerable a falsificaciones, pero debido a que NTP no mueve los relojes lejanos muy rápido, un reloj externo falso es raro que lo haga vulnerable a un ataque de repetición, aunque podría tener éxito en molestarlo haciendo que todos sus relojes se adelanten o atrasen.

1.4.10 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 permiten que las personas utilicen archivos remotos sin el gasto que representa transmitirlos nuevamente de un lado a otro y tratar de mantener versiones múltiples en sincronía. También es muy peligroso porque significa que permite a las personas leer sus datos sin obtener alguna autenticación adicional en la máquina donde estos residen. El sistema de archivos de RED (Network Filesystem) y el sistema de archivos Andrew (AFS) son los dos sistemas para archivos de redes en UNIX que se emplea con más frecuencia. NFS se diseñó para usarse en redes de área local y brinda respuestas rápidas, gran confiabilidad, sincronización de hora y un alto grado de confianza entre las 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 (Puede ser difícil) un atacante puede sencillamente montarlo en sus sistemas de archivos. En la forma en que funciona NFS las máquinas cliente tienen permitido leer y crear archivos guardados en el servidor sin tener que iniciar una sesión con este o teclear una contraseña. Debido a que NFS no registra las transacciones, usted tal vez ni siquiera que alguien más tiene acceso total a sus archivos.

NFS proporcionan una forma para controlar qué máquinas pueden tener acceso a los archivos. Un archivo llamado /etc/exports permite especificar que sistemas de archivos pueden montarse, y que 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 que máquinas puede montarlo, permite que cualquier máquina lo haga.

También son posibles varios ataques más sutiles a NFS. Por ejemplo, NFS tiene una autenticación de cliente muy débil y un atacante puede convencer al servidor NFS que una solicitud viene de un cliente que está permitido en el archivo exports. Asimismo, hay situaciones en las que un atacante puede robar un sistema de archivos que se encuentre ya montado.

Estos problemas se deben, en su mayoría al hecho de que NFS utiliza autenticación de Host, la cual es falsificada con facilidad. Debido a que NFS en realidad no funciona bien a través de Internet (supone una conexión más rápida entre hosts), no tienen caso permitirlo entre su sitio y esa red. Crea un problema de seguridad sin agregar funcionalidad.

AFS utiliza kerberos para dar autenticación y, opcionalmente la encriptación. Est5á diseñado para trabajar a través de redes de área amplia, incluyendo Internet. NFS se incluye como parte del sistema operativo con la mayoría de las versiones de UNIX; AFS es un producto adicional de otros fabricantes.

Debido a esto y a que AFS y Kerberos exigen mucha experiencia técnica para instalarse y mantenerse, AFS no se emplea masivamente fuera de un número pequeño de sitios grandes. Si tiene necesidad de hacer sistemas de archivos para red de área amplia seguros quizá valga la pena investigar AFS.

1.4.11 SISTEMAS DE VENTANAS.

La mayoría de las máquinas UNIX proporcionan 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 usuarios, el acceso de terminal remota se vuelve menos útil, necesita gráficas no sólo texto. X11 le proporciona gráficas remotas. Por desgracia, lo hace proporcionando acceso total a todas las capacidades que le da cuando 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:

1. Obtener descargas de pantallas. Copias de cualquier cosa que se muestren en pantallas del usuario.
2. Leer las teclas que oprime el usuario, puede incluir contraseña del usuario.
3. Inyectar pulsaciones de teclas. Se verán como si las hubiera oprimido el usuario.

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

1.4.12 SISTEMAS DE IMPRESIÓN.

Tanto lp, el sistema de impresión de System V, como lpr, el sistema de impresión de BSD (las 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 ineficiente de transferir datos a través de Internet. No hay razón para permitirlo. Si necesita imprimir en un sitio a través de Internet o permitir a otro sitio utilizar sus impresoras es posible instalar alias especiales de correo que imprimen el correo al recibirlo. Este es el método que muchas compañías utilizan aún a través de redes de área amplia privadas porque es bastante más confiable

CAPÍTULO 2

INTRODUCCIÓN A LOS FIREWALLS

2.1 ALGUNAS DEFINICIONES DE FIREWALLS.

Por desgracia, no existe una terminología totalmente consistente para las arquitecturas y componentes de los Firewalls. Diferentes personas usan términos en formas distintas o, peor aún conflictivas. Además, estos mismos términos tienen a veces otro significado en otros campos relacionados con las redes; las siguientes definiciones sirven para el contexto de Firewall.

FIREWALL:

Un componente o conjunto de componentes que restringen el acceso entre una red protegida e Internet o en otros conjuntos de redes.

HOST

Un sistema de computo conectado a una red.

HOST BASTIÓN.

Sistema de cómputo que debe ser altamente seguro porque es vulnerable a un ataque, por lo general debido a que está expuesto a Internet y es el punto principal de contacto para usuarios de redes internas. Obtiene su nombre de las proyecciones altamente fortificadas en los muros externos de los castillos medievales.

HOST CON DOBLE ACCESO.

Sistema de cómputo de uso general que tiene por lo menos dos interfaces de red.

PAQUETE.

Unidad fundamental de comunicación en Internet.

FILTRADO DE PAQUETE.

La acción que ejecuta un dispositivo para controlar de forma selectiva el flujo de datos hacia y desde una red. Para controlar de forma selectiva el flujo de datos hacia y desde una red. Los filtros permiten o bloquean los paquetes, en general mientras se enrutan en una red a otra (con mayor frecuencia desde Internet a una red interna y viceversa). Para realizar el filtrado de paquetes debe establecer un conjunto de reglas que especifiquen que tipos de paquetes (por poner un caso, aquellos que van o vienen de una dirección IP o de un puerto específico) van a permitirse y que tipos van a bloquearse. El filtrado de paquetes puede ocurrir en un router en un bridge o en un host individual. A veces se conoce como de protección (screening).

RED DE PERIMETRO

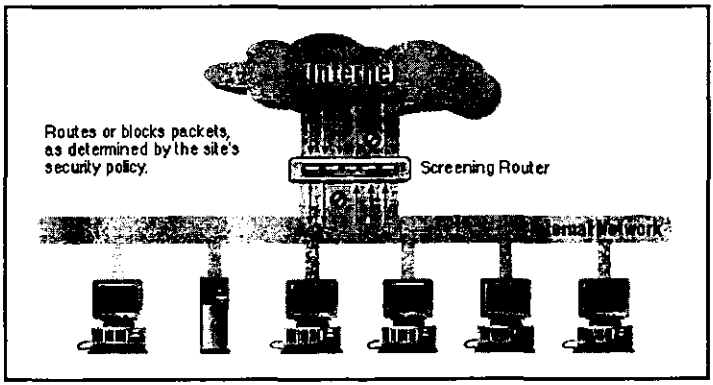
Es una red agregada entre una red de protección y una red externa a fin de poder proporcionar una capa adicional de seguridad. A una red de perímetro se le conoce a veces como VMZ (de las siglas De-Militarized Zone, zona desmilitarizada), por la zona que separa Corea del Norte a Corea del Sur.

SERVIDOR PROXY.

Programa que trata con servidores externos en nombre de cliente internos (son programas representantes). Los cliente proxy se comunican con los servidores proxy, los cuales, a su vez, transmiten solicitudes aprobadas de clientes a verdaderos servidores y de nuevo transmiten las respuestas a los clientes.

FILTRADO DE PAQUETES.

Los sistemas para filtrado de paquetes enrutan los paquetes entre hosts internos y externos, pero lo hacen en forma selectiva. Permiten o bloquean ciertos tipos de paquetes de una forma que refleja la propia política de seguridad de un sitio, como se muestra en (la figura). El tipo de router utilizado en un Firewall para filtrado de paquetes se conoce como router de protección (screening router).



Cada paquete tiene un conjunto de encabezados que contienen cierta información. La información principal es:

1. Dirección IP fuente.
2. Dirección IP destino.
3. Protocolo (si el paquete es TCP, UDP ó ICMP)
4. Puerto TCP, UDP ó ICMP fuente.
5. Puerto TCP, UDP ó ICMP destino.
6. Tipo de mensaje de ICMP.

Además, el router sabe información de los paquetes que no se reflejan en los encabezados, como:

1. La interface por la que llega el paquete.
2. La interface por la que sale el paquete.

El hecho que los servidores de servicios específicos de Internet residen en ciertos números de puertos permite que el router bloquee o permita ciertos tipos de conexiones con sólo especificar el número de puerto apropiado (por ejemplo, el puerto TCP 23 para conexiones Telnet) en el conjunto de reglas específicas para la filtración de paquetes.

A continuación se enumeran algunos ejemplos de formas en que puede programar un router de protección para enrutar paquetes de forma selectiva hacia o desde su sitio:

1. Bloquear todas las conexiones que entran de sistemas fuera de la red interna, excepto las conexiones SMTP entrantes (para que pueda recibir correo electrónico).

2. Bloquear todas las conexiones hacia o desde ciertos sistemas de los cuales desconfía.

3. Permitir correo electrónico y servicios FTP, pero bloquear servicios peligrosos como TFTP, el sistema Xwindow, RTC y los servidores "r" (rlogin, rsh, rcp, etc).

Para comprender como funciona un filtrado de paquetes, veamos la diferencia entre un router común y un router de protección.

Un router común sencillamente ve la dirección destino de cada paquete y selecciona la mejor forma que conoce para enviarlo a ella. La decisión sobre como manejar el paquete se basa sólo en un destino. Existen dos posibilidades: el router sabe como enviar el paquete a su destino y lo hace; el router no sabe como enviar el paquete a su destino y lo devuelve por medio de un mensaje de ICMP de "destino inalcanzable", a su dirección fuente.

Por otro lado, un router de protección ve más de cerca los paquetes. Además de determinar si puede o no enrutarlos a su destino, también determina si debe hacerlo o no. El "debe" o el "no debe" lo determina la política de seguridad del sitio; que el router de protección ha sido configurado para hacerla cumplir.

Aunque es posible colocar un sólo router de protección entre una red e Internet, esto asigna (como lo muestra la figura anterior) a el router una responsabilidad inmensa. No sólo debe hacer todo el enrutamiento y tomar las decisiones con relación este, sino que es el único sistema de protección, si falla su seguridad o se cae por un ataque, la red interna queda expuesta. Además, un router de protección directo no puede modificar los servicios. Puede permitir o negar un servicio, pero no puede proteger operaciones individuales dentro de un servicio. Si un servicio deseable tiene operaciones inseguras, o si el servicio es proporcionado por lo general con un servidor inseguro, el filtrado de paquetes por sí sólo no puede protegerlo.

Un sin número de otras arquitecturas han evolucionado para proporcionar seguridad adicional en implementaciones de Firewalls de filtrado de paquetes.

SERVICIOS PROXY

Los servicios proxy son programas de aplicación o servidores especializados que se ejecutan en un host Firewall: ya sea un host con doble acceso en la red interna y otra en la red externa, o algún host bastión que tiene acceso a Internet y que es accesible desde las máquinas internas. Estos programas toman las solicitudes de los usuarios para los servicios de Internet como FTP y Telnet y los envían, conforme sea apropiado con la política de seguridad del sitio, a los servicios reales. Los servidores proxy (o simplemente proxys) proporcionan conexiones sustitutas y actúan como compuertas a los servicios; por esta razón se conocen a veces como compuertas a nivel aplicación.

Los servicios proxy se encuentran, más o menos de manera transparente, entre un usuario (en la red interna) y un servicio externo (en Internet). En lugar de comunicarse entre sí de forma directa, cada uno lo hace con un proxy. Los servicios proxy manejan toda la comunicación entre usuarios y servicios de Internet de una forma transparente.

La transparencia es el principal beneficio de los servicios proxy. Son, en esencia, humo y espejos. Para el usuario, un servidor proxy presenta la ilusión de que trata directamente con el propio servidor. Al verdadero servidor, el servidor proxy presenta una ilusión de que trata directamente con un usuario en el host proxy (a diferencia del verdadero anfitrión del usuario).

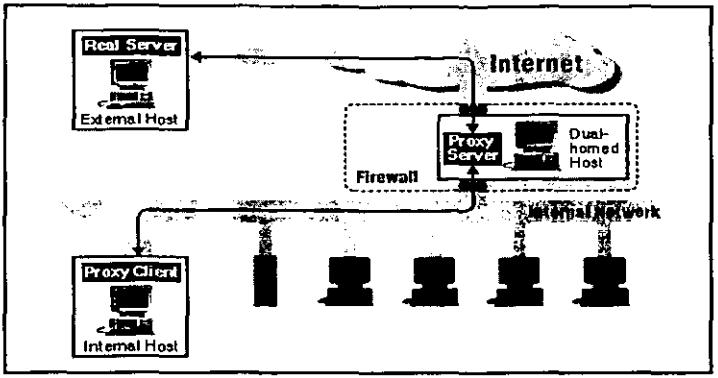
NOTA

Los servidores proxy son efectivos sólo cuando se emplean junto con un mecanismo que restringe las comunicaciones directas entre los hosts internos y externos. Los hosts con doble acceso y el filtrado de paquetes son un ejemplo de este tipo de mecanismo. Si los hosts internos pueden comunicarse en forma directa con hosts externos no hay necesidad de que los usuarios utilicen servicios proxy, así que no lo harán (en general). Es probable que eso no esté de acuerdo con su política de seguridad.

¿Cómo funcionan los servicios proxy? Veamos el caso más sencillo, donde los agregamos a un hosts con doble acceso.

Como muestra la siguiente figura, un servicio proxy requiere dos componentes: un servidor proxy y un cliente proxy. En esta situación el servidor proxy se ejecuta en el hosts con doble acceso. Un clinete proxy es una versión especial de un programa cliente común (por ejemplo, un cliente Telnet FTP) que se comunica con el servidor proxy en lugar de hacerlo con el "verdadero" servidor que esta en Internet; además, si a los usuarios se les enseñan procedimientos especiales que deben seguir, los programas clientes comunes con frecuencia pueden usarse como clientes proxy. El servidor proxy evalúa solicitudes del cliente proxy y decide cuales aprobar y cuales negar.

Se aprueba una solicitud el servidor proxy contacta al verdadero servidor en nombre del cliente (de ahí viene el termino proxy, representante), y procede a transmitir las solicitudes del cliente proxy al verdadero servidor, y la respuesta del verdadero servidor al cliente proxy.



En algunos sistemas proxy, en lugar de instalar software proxy cliente personalizado utilizará software estándar, pero para hacerlo configure procedimientos personalizados para usuarios.

Un servicio proxy es una solución de software no una arquitectura de Firewall en sí. Puede utilizar servicios proxy junto con cualquiera de las arquitecturas para Firewall.

El servidor proxy no siempre envía simplemente las solicitudes de los usuarios a los verdaderos servicios de Internet. También puede controlar lo que hacen los usuarios ya que pueden tomar decisiones sobre las solicitudes que procesa. Dependiendo de la política de seguridad de sitio, las solicitudes pueden permitirse o negarse.

Por ejemplo, el servicio proxy FTP podría negarse al dejar que los usuarios exporten archivos, o podría permitir que lo hagan sólo desde ciertos sitios. Los servicios proxy más refinados podrían permitir diferentes capacidades a distintos hosts, en lugar de poner las mismas restricciones a todos.

Existe excelente software disponible para servicios proxy. SOCKS es una herramienta para construcción de servicios proxy, diseñado para facilitar las conversiones cliente-servidor existente a versiones proxy de esas mismas aplicaciones. Trusted Information System Internet Firewall Toolkit (TIS FWTK) incluye servidores proxy para múltiples protocolos comunes de Internet incluyendo Telnet, FTP, HTTP, rlogin, x11 y otros; estos servidores proxy están diseñados para emplearse junto con procedimientos de usuario personalizados.

Muchos programas clientes y servidores estándar, tanto comerciales como gratuitos, ahora vienen equipados con sus propias capacidades de servicios proxy, ó con soporte para sistemas proxy genéricos como SOCKS. Esas capacidades pueden activarse al momento de su ejecución ó compilación.

2.1.1 USO DE UNA COMBINACIÓN DE TECNICAS Y TECNOLOGIAS

La "solución correcta" para construir un Firewall es, rara vez, una sola técnica: por lo general es una combinación de técnicas desarrolladas con astucia y cuidado para solucionar diferentes problemas. Los problemas que debe de resolver depende de los servicios que quiera proporcionar a sus usuarios y del nivel de riesgo que este dispuesto a aceptar. Qué técnicas utiliza para resolver esos problemas depende de cuanto tiempo, dinero y experiencia tenga.

Algunos protocolos (Telnet, FTP) se pueden manejar con más efectividad con filtrado de paquetes. Otros (FTP, Archie, Gopher, y WWW) se manejan con más efectividad con servidores proxy. La mayoría de los Firewalls emplean una combinación de servidores proxy y filtrado de paquetes.

2.2 ARQUIRECTURAS DE FIREWALLS

2.2.1 ARQUITECTURAS DE HOSTS CON DOBLE ACCESO.

Una arquitectura de host con doble acceso se construye alrededor de una computadora anfitrión que tiene por lo menos dos interfaces de red. Tal anfitrión podría actuar como un router entre las redes a las que están conectadas sus interfaces: es capaz de enrutar paquetes IP de una red a otra. Sin embargo, al implementar una arquitectura de Firewalls de tipo host con doble acceso, desactiva esta función de enrutamiento. Así, los paquetes IP de una red (por ejemplo, Internet) no se enrutan de forma directa con la otra red (la red interna protegida). Los sistemas dentro del Firewall puede comunicarse con el host con doble acceso, pero estos sistemas no pueden comunicarse entre sí de manera directa. El tráfico IP entre ellos está bloqueado en su totalidad.

La arquitectura de red para un Firewall de host con doble acceso es bastante sencilla: el host se coloca entre, y se conecta a, Internet y la red interna.

Los host con doble acceso pueden proporcionar un alto nivel de control. Si no se permite que los paquetes pasen entre redes externas e internas, puede tener la seguridad de que cualquier paquete en la red interna tenga una fuente externa es evidencia de algún tipo de problema de seguridad. En algunos casos, un host con doble acceso permitirá que rechace conexiones que pretenden ser para un servicio específico pero que en realidad, no contiene el tipo de datos correctos. (un sistema de filtrado de paquetes , por otro lado , tiene dificultades con este nivel de control). Sin embargo, es necesario mucho trabajo para aprovechar de manera consistente las ventajas potenciales de host con doble acceso.

Un host con doble acceso sólo puede proporcionar servicios de tipo proxy, o hacer que los usuarios inicien una sesión directa con él. Por ejemplo las cuentas de usuarios presentan problemas de seguridad importantes por sí mismas. Presentan problemas especiales en host con doble acceso, donde pueden, inesperadamente activar los servicios que usted considera inseguros. Además, la mayoría de los usuarios encuentran que es inconveniente emplear un host con doble acceso iniciando una sesión con él antes de salir a Internet.

Los servicios proxy son mucho menos problemáticos, pero que quizá no estén disponibles para todos los servicios que le interesen.

2.2.2 ARQUITECTURA DE HOST DE PROTECCIÓN.

Mientras una arquitectura de host con doble acceso proporcionar servicios desde un anfitrión conectado a varias redes (pero con el enrutamiento desactivado), una arquitectura de host de protección proporciona servicios en su servidor conectado sólo a la red interna, utilizando un router independiente. En esta arquitectura, la seguridad principal la proporciona el filtrado de paquetes (por ejemplo, el filtrado evita que las personas evadan los servidores proxy para hacer conexiones directas).

El host bastión está colocado en la red interna. El filtrado de paquetes en el router de protección está configurado de tal manera que el host bastión es el único sistema en la red interna con el que los servidores en Internet pueden abrir conexiones (por ejemplo, para entregar correo electrónico). Aún así, sólo están permitidas ciertos tipos de conexiones. Cualquier sistema externo que intente tener acceso a los sistemas o servicios internos tendrá que conectarse con este host. Por lo tanto, el host bastión debe mantener un alto nivel de seguridad.

El filtrado de paquetes también permite que el host bastión abra conexiones permitidas al mundo exterior (lo "permitido" lo determina la política de seguridad específica de su sitio).

La configuración para el filtrado de paquetes en el router de protección puede hacer una de las siguientes tareas:

1. Permitir que otros servidores internos abran conexiones con servidores en Internet para cientos servicios (permitiendo el servicio a través del filtrado de paquetes)

2. No permitir todas las conexiones de servidores internos (obligándolos a utilizar servicios proxy a través del host bastión)

Puede mezclar e igualar estos enfoques para diferentes servicios; algunos pueden permitirse directamente por el filtrado de paquetes, otros pueden permitirse de manera indirecta sólo por medio de proxies. Todo depende de la política específica que su sitio intente cumplir.

Debido a que esta arquitectura permite que los paquetes se muevan en Internet a las redes internas, puede parecer más riesgoso que la arquitectura de host con doble acceso, diseñado para que ningún paquete externo alcance la red interna. Sin embargo, en la práctica, la arquitectura de host con doble acceso es también propenso a fallas que permiten que, en realidad, pasen los paquetes de la red externa a la interna (debido a que esta clase de falla es inesperada, es poco probable que existan protecciones contra este tipo de ataques).

Además, es más fácil defender un router, que proporciona un conjunto muy limitado de servicios, que defender un host. para casi todos los propósitos, la arquitectura de host de protección proporciona mejor seguridad y mejor uso que la arquitectura de host con doble acceso.

Sin embargo, en comparación con otras arquitecturas, como la de subred de protección, tiene algunas desventajas. La principal es que si un atacante logra penetrar el host bastión, no queda nada en la ruta de seguridad de la red entre ese host y el resto de los hosts internos. El router también presenta un sólo punto de falla; si éste se haya en peligro, toda la red estará en merced de una atacante.

2.2.3 ARQUITECTURA DE SUBRED DE PROTECCIÓN.

La arquitectura de subred de protección agrega una capa adicional de seguridad a la arquitectura de host de protección al añadir una red de perímetro que aísla aún más la red interna de Internet.

¿Por qué hacer esto? Por su naturaleza, los host bastión son las máquinas más vulnerables de la red. A pesar de sus mejores esfuerzos por protegerlas, son las máquinas que más probablemente sean atacadas pues son las que pueden ser atacadas. Si, al igual, que una arquitectura de host de protección, su red interna está totalmente abierta para ser atacada desde su host bastión, entonces este es un blanco muy tentador. No hay alguna otra defensa entre él y sus máquinas internas (aparte de cualquier seguridad de host que puedan tener, que por lo general es muy poca). Si alguien entra al host bastión en una arquitectura de host de protección, se sacó la lotería.

Al aislar al host bastión en una red de perímetro, puede reducir el impacto de una entrada forzada a él. Ya no es una lotería garantizada; le da a el intruso un poco de acceso, pero no del todo.

Con la forma más sencilla de una arquitectura de subred de protección, hay dos routers de protección, cada uno conectado a la red de perímetro. Uno colocado entre a la red de perímetro y la red interna, y el otro entre la de perímetro y la red externa (por lo general, Internet). Para entrar a la red interna con este tipo de arquitectura, un atacante tendría que penetrar ambos routers. Aunque el atacante lograra de alguna forma penetrar al host bastión, aún tendría que pasar al router interior. No hay un punto vulnerable único que ponga en riesgo la red interna.

Algunos sitios van tan lejos como para crear una serie de capas de redes de perímetro entre el mundo exterior y su red interna. Los servicios menos confiables y más vulnerables se colocan en las redes de perímetros exteriores, más lejos de la red interior. La idea es que al atacante que penetre una máquina en la red de perímetro exterior, le costará más trabajo atacar con éxito las máquinas internas debido a las capas adicionales de seguridad entre el perímetro exterior y la red interna. Sin embargo, esto es sólo cierto si las distintas capas tienen sentido; si los sistemas de filtrado entre cada capa permiten lo mismo entre todas las capas, las capas adicionales no proporcionan mayor seguridad.

2.2.4 RED DE PERÍMETRO.

La red de perímetro es otra capa de seguridad, una red adicional entre la red externa y la interna protegida. Si un atacante entra con éxito a los límites externos del Firewall, la red de perímetro ofrece una capa adicional de protección entre el atacante y los sistemas internos.

Un ejemplo de por que una red puede ser útil se presente así. En muchas configuraciones de red, es posible para cualquier máquina en una red determinada monitorear el tráfico de cada máquina de la red. Esto es cierto para la mayoría de redes basadas en ethernet (ethernet es, por mucho, la tecnología para redes LAN más común hoy en día); esto es también válido para varias otras tecnologías populares, como Token Ring y FDDI. Los atacantes pueden tener éxito en saber contraseñas monitoreando las que utilizan durante las sesiones de Telnet, FTP y rlogin. Aunque las contraseñas no estén en riesgo, los atacantes aun pueden ver rápidamente el contenido de los archivos delicados que las personas pueden estar usando, correo electrónico interesante que pueden estar leyendo, etc.; el atacante puede, en esencia, "ver por encima del hombro" de cualquier persona que use la red.

Con una red de perímetro, si alguien entra a un host bastión, podrá curiosear sólo su tráfico. Todo el tráfico en la red de perímetro debe ser de o para el host bastión, o hacia o desde Internet.

Debido a que el tráfico estrictamente interno (esto es, tráfico entre dos servidores internos que se supone es importante o privado) no pasa por la red de perímetro, estará seguro de la vista de los atacantes si el host bastión está en peligro.

Es obvio que el tráfico que viene o va hacia el host bastión, el mundo externo todavía estará visible.

2.2.5 HOST BASTIÓN.

Con la arquitectura de subred de protección, usted enlaza un host bastión (o hosts) a la red de perímetro; este hosts es el principal punto de contacto para las conexiones que entren desde el mundo exterior; por ejemplo:

1. Sesiones de correo electrónico (SMTP) para entrega de mensajes al sitio.

2. Conexiones FTP que entran al servidor FTP anónimo del sitio.

3. Consultas que entran al servicio de nombres de dominio (DNS) del sitio, etc.

Los servicios que salen (de clientes internos a servidores en Internet) se manejan de cualquiera de estas formas:

1. Configurar el filtrado de paquetes en los routers exteriores e interiores a fin de permitir que los clientes internos tengan acceso a los servidores externos de forma directa.

2. Instalar los servidores proxy para que se ejecuten en el host bastión (si su Firewall emplea software proxy) para permitir a los clientes internos tener acceso a los servidores externos de manera directa. También configurar el filtrado de paquetes a fin de permitir a los clientes internos comunicarse con los servidores proxy en el host bastión y viceversa, pero prohibir las comunicaciones directas entre clientes internos y el mundo exterior.

En cualquier caso, el filtrado de paquetes permite al host bastión conectarse a, y aceptar conexiones de, servidores en Internet; qué servidores y para qué servicios lo dicta la política de seguridad de su sitio.

La mayoría de las tareas del host bastión es actuar como servidor proxy para varios servicios, ya sea ejecutando software de servidor proxy especializado para protocolos específicos (como HTTP o FTP), ejecutando servidores estándar para protocolos basados en proxy (como SMTP).

2.2.6 ROUTER INTERIOR.

El router interior (a veces llamado Router de choque) protege la red interna tanto de Internet como la red de perímetro.

El router interior realiza la mayor parte del filtrado de paquetes para el Firewall. Permite que varios servicios seleccionados salgan de la red interna hacia Internet. Estos servicios son los que su sito puede soportar y proporcionar seguridad utilizando el filtrado de paquetes en lugar de servicios proxy. (Su sitio debe establecer su propia definición de qué significa "seguro". Tendrá que considerar sus propias necesidades, capacidades y limitaciones; no hay una respuesta única para todos los sitios). Los servicios que podría permitirse incluyen Telnet, FTP, WAIS, Archie, Gopher y otros, conforme sean apropiados para sus necesidades y preocupaciones.

Los servicios que permite el router interior en su host bastión (en la red de perímetro) y su red interna no son necesariamente los mismo que el mismo router permite entre Internet y su red interna. El objetivo de limitar los servicios entre el host bastión y su red interna es reducir el número de máquinas (y de servicios de esas máquinas) que pueden ser atacadas desde el anfitrión bastión, en caso de verse comprometido.

Debe limitar los servicios permitidos entre el host bastión y la red interna sólo a los que en realidad necesite, como SMTP (para que el host bastión pueda enviar el correo electrónico que entra), DNS (para que el host bastión pueda responder a las preguntas de las máquinas internas, o preguntarles, dependiendo de su configuración), etc. Debe limitar aún más los servicios, hasta donde sea posible, permitiéndoles ir o venir de host internos determinados; por ejemplo, SMTP podría estar limitado sólo a conexiones entre el host bastión y su servidor o servidores de correo interno.

2.2.7 ROUTER EXTERIOR.

En teoría, el router exterior (a veces llamado router de acceso) protege tanto la red de perímetro como la red interna de Internet. en la práctica, los routers exteriores tienden a permitir cualquier cosa que salga de la red de perímetro y, en general, realizan muy poco filtrado de paquetes. Las reglas para el filtrado para proteger la máquinas internas tienen que ser, en esencia, las mismas tanto para el router interior como para el exterior; si hay una error en las reglas que permitan el acceso a un atacante, es probable que esté presente en ambos routers.

Con frecuencia, un grupo externo (por ejemplo, su proveedor de Internet) proporciona el router, y su acceso a él puede ser limitado. Un grupo externo que mantiene un router quizá esté dispuesto a establecer algunas reglas generales para el filtrado de paquetes, pero no querrá mantener un conjunto de reglas complicadas o que cambien con frecuencia.

Asimismo, puede no confiar tanto en ellos como en sus propios routers. Si el router se descompone e instalan uno nuevo, ¿se acordarán de configurar nuevamente los filtros? ¿acaso se van a molestar en mencionar que reemplazaron el router para que usted lo sepa y lo revise?.

Las únicas reglas para el filtrado de paquetes realmente especiales en el router extrior son las que protegen las máquinas de la red de perímetro (esto es, los hosts bastión y el router interno). Sin embargo, en general no se necesita mucha protección porque los servidores en la red de perímetro están protegidos, principalmente, a través de la seguridad del host (aunque nunca esté de más la redundancia).

El resto de las reglas que podría establecer en el router exterior son duplicados de las del router interior. Son reglas que evitan que el tráfico inseguro pase entre los hosts internos e Internet. Para soportar los servicios proxy, donde el router interior permita que los host internos envíen algunos protocolos siempre y cuando hablen con el host bastión, el router exterior podría dejar pasar esos protocolos siempre y cuando vengan del host bastión. Estas reglas son deseables para dar un nivel adicional de seguridad, pero en teoría sólo bloquean paquetes que no pueden existir porque ya han sido bloqueados por el router interior. Si en realidad existen, es porque el router interior falló o alguien ha conectado un host inesperado en la red de perímetro.

Entonces, ¿qué necesita hacer, en realidad, el router exterior? Una de las tareas de seguridad que puede realizar con éxito (y que no se puede hacer fácilmente en algún otro lugar) es el bloqueo de cualquier paquete que entra de Internet que tiene direcciones fuentes falsificadas. Tales paquetes dicen venir desde la red interna, pero en realidad entrena de Internet.

El router interior puede hacer esto, pero no determinar si los paquetes que dicen provenir de la red de perímetro son falsificados. Aunque la red de perímetro no debe tener nada totalmente confiable en ella, de todas formas debe ser más confiable que el universo exterior, poder falsificar paquetes desde ella le da al atacante casi todas las ventajas para comprometer al host bastión. El router exterior se encuentra en un límite más claro. El router interior tampoco puede proteger los sistemas que están en la red de perímetro contra paquetes falsificados.

2.3 VARIACIONES EN LAS ARQUITECTURAS DE FIREWALLS

Hay una gran flexibilidad en la configuración y combinación de los componentes de un Firewall para adaptarse mejor a su hardware, su presupuesto y su política de seguridad.

2.3.1 ES CORRECTO UTILIZAR MÚLTIPLES HOSTS BASTIÓN.

Quizá decida hacer que host bastión maneje los servicios importantes para sus usuarios (como servidores SMTP, servidores proxy, etc), mientras otro host maneja los servicios que proporciona a Internet pero que no le importan a sus usuarios (por ejemplo, un servidor FTP anónimo). De esta forma, el rendimiento de sus propios usuarios no se verá mermado por las actividades de sus usuarios externos.

También es posible proporcionar múltiples host bastión con los mismos servicios por razones de rendimiento, pero puede ser difícil lograr el equilibrio en la carga. La mayoría de los servicios deben estar configurados para servidores determinados, así que crear varios hosts para servicios individuales funciona mejor si puede predecir oportunamente su uso.

¿Qué hay sobre la redundancia? Si la configuración de su Firewall incluye varios hosts bastión, quizá lo configure para obtener redundancia; de este modo, si uno falla, los servicios puedan ser proporcionados por otro, pero debe tener cuidado porque sólo algunos servicios soportan este enfoque. Por ejemplo, podría configurar y designar varios hosts bastión como servidores DNS para su dominio (por medio de registros NS de DNS que especifican los servidores de nombre para su dominio), o como servidores SMTP (por medio de registros MX {Intercambio de correo} de DNS, que especifican qué servidores aceptarán correo para un host o dominio determinado), o ambos. Así, si uno de los hosts bastión no está disponible o está sobrecargado, la actividad de DNS y SMTP utilizará el otro como un sistema de apoyo.

2.3.2 ES CORRECTO FUSIONAR EL ROUTER INTERIOR Y EXTERIOR.

Puede fusionar los routers interior y exterior en uno sólo, pero únicamente si tiene un router con suficiente capacidad y flexibilidad. En general, necesita un router que permita especificar tanto los filtros de salida como los de entrada en cada interface.

Si fusiona los routers interior y exterior, aún tendrá la red de perímetro (en una interface del router) y una conexión a su red interna (para otra interface del router). Cierta parte del tráfico fluirá de manera directa entre la red interna e Internet (el tráfico permitido por las reglas establecidas en el router para el filtrado de paquetes) y la otra parte fluirá entre la red de perímetro e Internet, o la red de perímetro y la red interna (el tráfico manejado por los servidores proxy).

Esta arquitectura, como la de host de protección, hace vulnerable el sitio porque compromete un solo router. En general, los routers son más fáciles de proteger que los hosts, pero no son impenetrables.

2.3.3 ES CORRECTO FUSIONAR EL HOST BASTION Y EL ROUTER EXTERIOR.

Quizá haya casos en que utilice una sola máquina con doble acceso como host bastión y como router exterior. He aquí un ejemplo: supongamos que sólo tiene una conexión SLIP o PPP conmutada con Internet. En este caso, podría ejecutar algo como el paquete Morning Star PPP en su host bastión, y dejarlo actuar como un host bastión y como router exterior. Esto equivale, en función, a la configuración de tres máquinas (host bastión, router interior y router exterior) descrita para la arquitectura de subred de protección.

A diferencia de fusionar los routers interior y exterior, fusionar el host bastión con el router exterior, no crea nuevas vulnerabilidades significativas. Lo que sí hace es exponer aún más el host bastión. En esta arquitectura el host bastión está más expuesto a Internet, protegido sólo por cualquier filtrado (si lo hay) que haga su propio paquete de interface, y necesita de cuidados adicionales para protegerlo.

2.3.4 ES PELIGROSO UTILIZAR MÚLTIPLES ROUTERS INTERIORES.

Usar varios routers interiores para conectar su red de perímetro a varias partes de la red interna puede causar muchos problemas y, en general, es mala idea.

El problema básico es que el software de enrutamiento en un sistema interno podría decidir que la vía más rápida hacia otro sistema interno es a través de la red de perímetro. Si tiene suerte, este enfoque simplemente no funcionara porque estará bloqueado por el filtrado de paquetes de uno de los routers. Si no tiene suerte, funcionará y tendrá tráfico sensible, totalmente interno, fluyendo a través de la red de perímetro, donde puede ser observado si alguien logra entrar al host bastión.

También es difícil mantener correctamente configurados varios routers internos. El router interior es el que tiene el conjunto más importante y complejo de filtros de paquetes, y tener dos de ellos duplica sus oportunidades de que queden mal establecidas las reglas.

En una red interna grande, tener un sólo router interno puede constituir un problema de rendimiento y un problema de confiabilidad. Si intenta proporcionar redundancia, ese solo punto de falla es una molestia mayor. En tal caso, lo más seguro (y redundante) es instalar cada router interior a una red de perímetro y aun router exterior independientes.

En la mayoría de los casos, router interno no es el cuello de botella del rendimiento. Si lo es, entonces ocurre alguno de los siguientes casos:

1. Hay mucho tráfico que va de la red de perímetro y que no pasa; por lo tanto a la red externa.
2. Su compuerta exterior es más rápida que su compuerta interior.

En el primer caso se ha configurado mal; la red de perímetro puede captar tráfico ocasional que no está destinado al mundo exterior en ciertas configuraciones (por ejemplo, las consultas DNS sobre los hosts externos cuando la información es almacenada en memoria caché), pero este tráfico nunca debe ser significativo. En el segundo caso, debe considerar seriamente actualizar el router interior para igualar al router exterior en lugar de agregar otro.

Otra razón para instalar routers interiores es que tiene varias redes internas, las cuales tienen razones técnicas, organizativas o políticas para no compartir un sólo router. La forma más sencilla de acomodar estas redes sería darles interfaces separadas en un sólo router, como se muestra en la figura siguiente, lo cual complica bastante la configuración del router, pero no produce los riesgos de una configuración de múltiples routers interiores. Si hay demasiadas redes para un sólo router, o si compartir un router es inaceptable por otras razones, considere poner un backbone interno (red dorsal interna) y conectarlo a la red de perímetro con un sólo router.

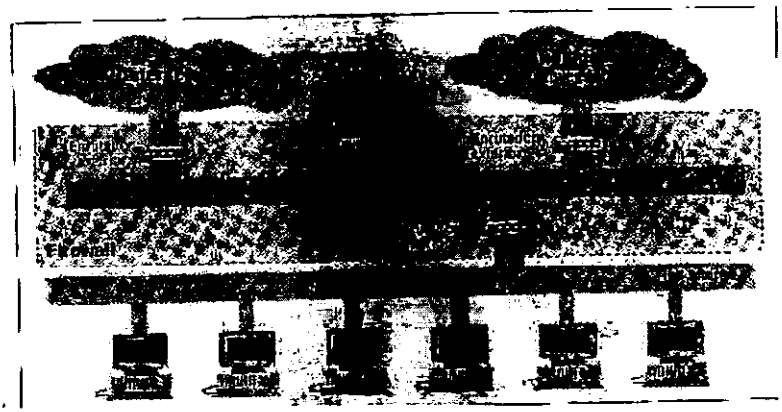
Quizá parezca que una forma efectiva de adecuar diferentes políticas de seguridad entre distintas redes internas es conectarlas al perímetro a través de routers independientes (por ejemplo, una red quiera permitir conexiones que otros consideren inseguras). En este caso la red de perímetro debe ser la única interconexión entre las redes internas; no debe pasar tráfico confidencial entre ellas; y cada red interna debe tratar a la otra como una red externa no confiable.

Es posible que esto resulte muy inconveniente para algunos usuarios de cada red, pero cualquier otra cosa comprometerá la seguridad del sitio por completo o suprimirá la distinción que lo llevó a instalar dos routers.

Si decide aceptar los riesgos de tener múltiples routers internos, puede minimizarlos haciendo que el mismo grupo los maneje todos (para que no se hagan cumplir las políticas de seguridad conflictivas). También debe observar con cuidado que no pase tráfico interno por la red de perímetro y actuar con prontitud para solucionar su causa en caso que se presente esta situación.

2.3.5 ES CORRECTO UTILIZAR MÚLTIPLES ROUTERS EXTERIORES.

Existen ciertos casos en que tiene sentido conectar múltiples routers exteriores a la misma red de perímetro, como lo muestra la siguiente figura:



Los ejemplos son:

1. Tiene varias conexiones con Internet (por ejemplo, a través de diferentes proveedores de servicio, para tener redundancia).
2. Tiene conexión a Internet, más otras conexiones a otros sitios.

En estos casos, resultaría mejor tener un router exterior con múltiples interfaces de red exterior.

Anexar varios routers exteriores que van a la misma red externa (por ejemplo, dos proveedores de Internet distintos) no es un problema de seguridad significativo. Pueden tener diferentes conjuntos de filtros, pero eso no es vital en los routers exteriores. Existe el doble de posibilidades de que uno sea comprometido, pero que eso no ocurra no es muy amenazante.

Las cosas son más complejas si las conexiones son a diferentes lugares (por ejemplo, una a Internet y otra a un sitio con el que está colaborando y para el cual necesita más ancho de banda). Para saber si una arquitectura de este tipo tiene sentido en estos casos, debe hacerse está pregunta ¿qué tráfico podrían ver si entran al host bastión en esta red de perímetro? Por ejemplo, si entrara un atacante, ¿podría curiosear sensible entre su sitio y un subsidiario o afiliado? En caso de ser así, entonces quizá deba pensar en instalar varias redes de perímetro.

2.3.6 ES CORRECTO TENER MÚLTIPLES REDES DE PERÍMETRO.

Existen ciertas situaciones en las que tiene sentido que la configuración incluya múltiples redes de perímetro.

Podría poner varias redes de perímetro para proporcionar redundancia. No tiene mucho sentido pagar por dos conexiones a Internet y luego ejecutar ambas a través del mismo router o routers. Poner dos routers exteriores, dos redes de perímetro y dos routers interiores asegura que no haya un sólo punto de falla entre nuestra red e Internet.

También se podrían colocar múltiples redes de perímetro para su privacía, a fin de transmitir datos moderadamente confidenciales a través de una, y una conexión a Internet a través de la otra. En este caso, hasta puede conectar ambas redes de perímetro al mismo router interior.

Tener múltiples redes de perímetro es menso riesgoso que tener múltiples routers interiores compartiendo la misma red interna, pero aún así el mantenimiento es un problema. Es probable que tenga múltiples routers internos, presentando varios posibles puntos de riesgo. Esos routers deben vigilarse con cuidado para que continúen aplicando políticas de seguridad apropiadas; si ambos se conectan a Internet, deben aplicar la misma política.

2.3.7 ES CORRECTO UTILIZAR HOST CON DOBLE ACCESO Y SUBREDES DE PROTECCIÓN.

Se pueden lograr incrementos importantes en la seguridad al combinar una arquitectura de host con doble acceso con una arquitectura de subred de protección. Para hacer esto, divida la red de perímetro y conecte un host con doble acceso. Los routers proporcionan protección contra la falsificación y protegen de fallas en donde el host con doble acceso empieza a enrutar tráfico.

Este host proporciona controles más finos en las conexiones de filtrado de paquetes. Este Firewall es de cinturón y tirantes, proporciona una excelente protección de varias capas, aunque requiere de una configuración cuidadosa en el host con doble acceso para asegurar que le brinde plenas ventajas de las posibilidades.

3.1 HOST BASTIÓN

El host bastión es su presencia pública en Internet. Piense en él como en el vestíbulo de un edificio. Los extraños quizá no puedan subir las escaleras y quizá no puedan entrar a los elevadores, pero pueden entrar libremente al vestíbulo y preguntar por lo que quieren (si obtienen o no lo que pidieron, depende de la política de seguridad del edificio). Al igual que el vestíbulo del edificio, host bastión está expuesto a los elementos potencialmente hostiles. El host bastión es el sistema al que los extraños (amigos o posibles enemigos) deben conectarse por lo regular para tener acceso a un sistema o servicio que está dentro del Firewall.

Por diseño, el host bastión está muy expuesto, ya que su existencia es conocida en Internet. Por está razón, los constructores y administradores de Firewalls necesitan concentrar los esfuerzos de seguridad en él. Deben poner atención especial en el host durante la construcción inicial y operación continua. Debido a que el host bastión es el más expuesto, también debe ser el más resguardado.

Los principios y procedimientos para construir un host bastión son extensiones de los que se utilizan para asegurar cualquier host.

3.1.1 PRINCIPIOS GENERALES.

Existen dos principios básicos para diseñar y construir un host bastión: manténgalo sencillo y prepárese para cuando esté comprometido.

Manténgalo sencillo.

Entre más sencillo sea su host bastión, es más fácil de asegurarlo.

Cualquier servicio que ofrece el host bastión podría tener problemas de software o errores en la configuración, y cualquiera de éstos pueden ocasionar problemas de seguridad. Por lo tanto, querrá que el host bastión haga lo menos posible. Debe proporcionar el conjunto más pequeño de servicios con los menores privilegios posibles, pero que continúe cumpliendo con su tarea.

Esté preparado para cuando el host bastión esté comprometido.

A pesar de sus mejores esfuerzos para procurar la seguridad del host bastión, pueden ocurrir entradas forzadas. No sea ingenuo. Sólo anticipando lo peor y haciendo planes para ello, tendrá más posibilidades de impedirlo. Siempre tenga presente esta pregunta: "¿Qué pasa si está comprometido el host bastión?" mientras lleva a cabo los pasos necesarios para asegurarlo, al igual que el resto de la red.

¿Por qué subrayamos este punto? La razón es sencilla: el host bastión es la máquina más lógica de ser atacada porque es la más accesible al mundo exterior. También es la máquina de la cual es más probable que vengan los ataques contra sus sistemas internos, porque es posible que el mundo externo no pueda comunicarse con sus sistemas internos de manera directa. Haga el mejor esfuerzo para asegurar que no entren al host bastión, pero tenga presente esta pregunta: ¿y si lo hacen?

En caso de que entren al host bastión, no querrá que esta entrada lleve a que se comprometa todo el Firewall. Puede evitarlo no dejando que las máquinas internas confíen más en el host bastión de lo que sea absolutamente necesario para que funcione. Debe ver con cuidado cada servicio que proporciona el host bastión a las máquinas internas, y determinar, con base en cada servicio, qué tanta confianza y privilegios debe tener, en realidad, cada servicio.

Una vez tomadas estas decisiones, puede utilizar varios mecanismos para hacerlas cumplir. Por ejemplo, puede instalar mecanismos estándar para el control de acceso (contraseñas, dispositivos de autenticación, etc.) en los hosts internos, o puede instalar un filtrado de paquetes entre el host bastión y los host internos.

3.1.2 TIPOS ESPECIALES DE HOSTS BASTIÓN.

Host con doble acceso sin enrutamiento.

Un host con doble acceso sin enrutamiento tiene varias conexiones de red, pero no pasa el tráfico entre ellas. Un host así podría ser un Firewall por sí solo, o podría ser parte de un Firewall más complejo. En general, los hosts con doble acceso sin enrutamiento están configurados como otros host bastión, pero necesitan precauciones adicionales, para asegurarse de que en realidad no ofrezcan enrutamiento. Si un host de este tipo es todo su Firewall, debe ser muy pero muy obstinado en su configuración y seguir con cuidado extremo las instrucciones habituales para el host bastión.

Máquinas víctimas.

Quizá querrá ejecutar servicios difíciles de proporcionar de manera segura o servicios que son tan nuevos que no sabe cuáles son las repercusiones en la seguridad por medio de servicios proxy o con el filtrado de paquetes. Para ese propósito, una máquina víctima (o chivo expiatorio) puede ser útil. Se trata de una máquina que no tiene nada de importante y que no tiene acceso a las máquinas de las cuales se podría aprovechar un intruso. Proporcionar sólo lo absolutamente necesario para ser utilizado por los servicios que usted necesita. Si es posible, proporciona sólo un servicio que no es seguro o que no ha sido probado, para evitar interacciones inesperadas.

Hosts bastión internos

En la mayoría de las configuraciones, el host bastión principal tiene interacciones especiales con ciertos hosts internos. Por ejemplo, quizá pase correo electrónico a un servidor de correo interno, coordinando un servidor interno de nombres, o pasando noticias de Usenet a un servidor interno de noticias. Éstas máquinas son, en realidad, hosts bastión secundarios y deben configurarse y protegerse más como host bastión que como hosts internos normales. Quizá deba dejar activados más servicios en ellos, pero debe seguir el mismo proceso de configuración.

3.1.3 CÓMO SELECCIONAR UNA MÁQUINA.

El primer paso para construir un host bastión es decidir qué tipo de máquina utilizar. Querrá confiabilidad (si el host bastión se descompone, pierde la mayoría de los beneficios de su conexión a Internet), soporte y configuración.

Qué sistema operativo.

El host bastión debe ser algo con lo que esté familiarizado. Terminará personalizando la máquina y el sistema operativo extensamente. Debido a que un host bastión totalmente configurado es un ambiente muy restringido, querrá desarrollarlo en otra máquina, y ayuda mucho intercambiar sus periféricos con otras máquinas.

Es posible que no pueda proporcionar o dar acceso a todos los servicios que desea a través de su plataforma original, porque las herramientas relevantes (servidores proxy, sistemas de filtrado de paquetes y aun servidores normales para los servicios básicos, como SMTP y DNS) podrían no estar disponibles para esa plataforma.

UNIX es el sistema operativo más popular para ofrecer servicios de Internet, y las herramientas están ampliamente disponibles para construir hosts bastión en sistemas UNIX. Si no tiene plataformas apropiadas para un host bastión y de todas formas debe aprender un sistema operativo nuevo, se recomienda que se pruebe con UNIX porque ahí es donde encontrará el conjunto de herramientas más grande y extenso para construir hosts bastión.

Qué tan rápida debe ser una máquina.

El host bastión no necesita ser una máquina rápida; de hecho, es mejor que no sea muy potente. Existen varias buenas razones, además del costo, para hacer su host tan potente como sea necesario a fin de que cumpla con su trabajo, pero no más. No necesita muchos caballos de fuerza para proporcionar los servicios requeridos del host bastión.

El host bastión en realidad no tiene mucho que hacer. Lo que debe hacer está más bien limitado por la velocidad de su conexión con el mundo exterior, no sólo por la velocidad del cpu del host bastión en sí. No se necesita un procesador tan potente para manejar correo, DNS, FTP y servicios proxy.

Quizá necesite más potencia para soportar las solicitudes de Internet si su sitio se vuelve demasiado popular. Una compañía grande con varias conexiones Internet y servicios populares tal vez necesite utilizar hosts bastión así como máquinas grandes y potentes.

Qué configuración de hardware.

Si se quiere una configuración de hardware confiable, debe seleccionar una máquina base y dispositivos que no sean lo más nuevo en el mercado. También que la configuración cuente con soporte, así que no se seleccione algo tan viejo que no pueda encontrar "refacciones".

Aunque no necesita potencia pura en el CPU, sí requiere de una máquina que dé seguimiento a varias conexiones simultáneamente; esto es, un uso intensivo de memoria, así que querrá una gran cantidad de memoria y tal vez una gran cantidad de espacio de intercambio en disco (swap) también. Los servicios proxy-caché también necesitan una gran cantidad de espacio libre en el disco para utilizarse como caché.

El host bastión no necesita de gráficas interesantes, y no las debe tener. Es host para servicios de red; nadie necesita verlo.

3.1.4 CÓMO SELECCIONAR UNA UBICACIÓN FÍSICA.

El host bastión debe estar en una ubicación que sea segura físicamente. Hay dos razones para estos:

1. Es imposible asegurar adecuadamente una máquina contra un atacante que tiene acceso físico a ella; hay demasiadas formas de que el atacante puede comprometerla.

2. El host 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, perderá acceso a por lo menos algunos de los servicios.

Nunca subestime el poder la insensatez humana. Aunque no crea que valga la pena para alguien intervenir tiempo y esfuerzo para obtener acceso físico a la máquina a fin de entrar en ella, asegúrela para evitar que personas bien intencionadas dentro de su organización inadvertidamente la hagan insegura o no funcional.

Su anfitrión bastión debe estar en una habitación cerrada, con aire acondicionado y ventilación adecuados. Si proporciona energía continua para su conexión a Internet, asegúrese de proporcionarla también para el host bastión.

3.1.5 CÓMO UBICAR EL HOST BASTIÓN EN LA RED.

El host bastión debe localizarse en una red que no lleve tráfico confidencial, de preferencia una red especial para este fin.

La mayoría de las interfaces Ethernet y Token Ring puede operar en "modo promiscuo". En este modo, puede capturar todos los paquetes en la red a la que están conectadas, en lugar de sólo los paquetes dirigidos a esa máquina específica que es parte de la interface. Otros tipos de interfaces de red, como FDDI, tal vez no pueden capturar todos los paquetes, pero dependiendo de la arquitectura de la red, por lo común pueden capturar por lo menos algunos paquetes que no están específicamente dirigidos a ellas.

Esta capacidad tiene un propósito útil: para análisis, prueba y depuración de redes, por ejemplo, programas como etherfind y tcpdump. Por desgracia, también puede emplearse por un intruso para ver todo el tráfico en un segmento de red. Este tráfico podría incluir Telnet, FTP o sesiones rlogin (desde donde se pueden capturar claves de acceso y contraseñas), correo electrónico confidencial, accesos NFS de archivos sensibles, etc. Debe suponer lo peor: puede comprometerse el anfitrión bastión. Si se compromete, no querrá que el host bastión espíe este tráfico.

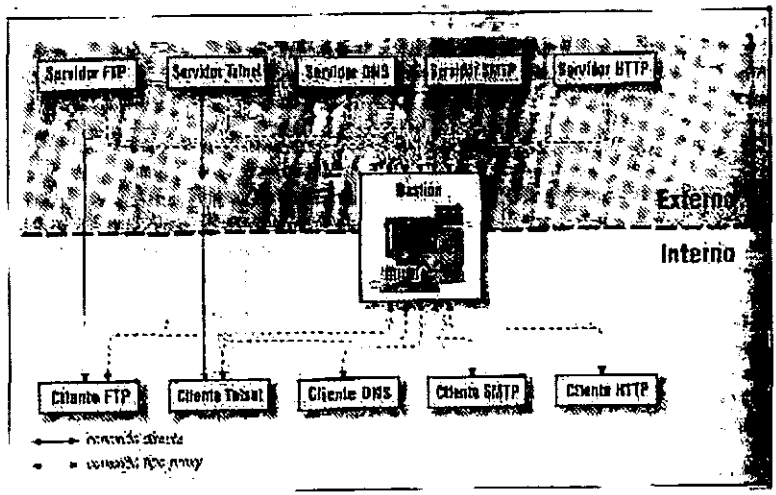
Una forma de enfocar el problema es no colocar el host bastión en una red interna: en lugar de eso, póngalo en una red de perímetro. Una red de perímetro es una capa adicional de seguridad entre su red interna e Internet. La red de perímetro está separada de la red interna por un router o un bridge. El tráfico interno se mantiene en la red interna y no está visible en la red de perímetro. Todo lo que puede ver el host bastión en la red de perímetro son paquetes que son de o hacia sí mismo, o desde o hacia Internet. Aunque este tráfico todavía podría ser algo delicado, tal vez lo sea mucho menos que el tráfico común de su red interna, y hay otros lugares (por ejemplo, su proveedor de servicios de Internet) que pueden ver mucho de él.

Usar la red de perímetro con un router para el filtrado de paquetes entre ella y la red interna le proporciona algunas ventajas adicionales. Limita aún más su exposición si el host bastión está comprometido, al reducir el número de host y servicios a los que el host bastión puede dar acceso.

Si no puede poner el host bastión en una red de perímetro, puede evaluar ponerlo en una red que no sea susceptible de ser espiada. Por ejemplo, podría ponerlo en un concentrador inteligente 10baseT, en un switch Ethernet o una ATM. Si sólo hace esto, debe tener cuidado adicional para asegurarse de que nada le inspire confianza al host bastión, porque no hay una capa adicional de protección entre él y la red interna. El uso de este tipo de tecnología para su red de perímetro es lo mejor de ambos mundos; el host bastión está aislado de los sistemas internos (como en una red de perímetro tradicional) pero no puede espiar el tráfico que circula en la red de perímetro.

3.1.6 COMO SELECCIONAR LOS SERVICIOS PROPORCIONADOS POR EL HOST BASTIÓN.

El host bastión proporciona cualquier servicio que necesite su sitio para tener acceso a Internet, o que quiere ofrecer a Internet (servicios que no se siente seguro de proporcionar en forma directa a través del filtrado de paquetes) (la figura muestra un conjunto típico) No debe poner ningún servicio en el host bastión que no tenga la intención de utilizar hacia o desde Internet. Por ejemplo, no debe proporcionar servicios de arranque para host internos (a no ser que, por alguna razón, tenga la intención de proporcionar servicios de arranque para hosts de Internet). Debe suponer que el host bastión estará comprometido y que todos los servicios que están en él estarán disponibles para Internet.



Puede dividir los servicios en cuatro clases:

Servicios que son seguros.

Los servicios de esta categoría pueden proporcionarse a través del filtrado de paquetes, si utiliza este enfoque (en un Firewall basado totalmente en servicios proxy, todo debe proporcionarse en el host bastión o no debe proporcionarse).

Servicios inseguros por cómo se proporcionan comúnmente pero que deben asegurarse

Los servicios de esta categoría pueden proporcionarse en el host bastión.

Servicios inseguros por cómo se proporcionan comúnmente y que no pueden asegurarse.

Éstos tendrán que desactivarse y proporcionarse en un host víctima si los necesita de verdad.

Servicios que no utiliza o que no utiliza junto con Internet.

De desactivar los servicios de esta categoría.

El correo electrónico (SMTP) es el más básico de los servicios que proporcionan los hosts bastión en general. Quizá también quiera dar acceso o proporcionar servicios de información como:

1. FTP - Transferencia de archivos.

2. Gopher - obtención de información basado en menús.

3. WAIS - obtención de información por medio de búsqueda de palabras clave.

4. HTTP - obtención de información por medio de hipertexto (WWW).

5. NTTP - servicios de Usenet.

Para soportar cualquiera de estos servicios (incluyendo de SMTP), tiene que dar acceso y proporcionar el Servicios de Nombres de Dominio (DNS). Rara vez se utiliza DNS directamente, pero soporta a todos los demás protocolos proporcionando la forma de traducir nombres de hosts a direcciones IP y viceversa.

3.1.7 NO PERMITA CUENTAS DE USUARIO EL HOST BASTIÓN.

Si es posible, no permita cuentas de usuario en el host bastión. Mantener esas cuentas alejadas de éste le dará la mejor seguridad. Hay varias razones, entre las que se cuentan:

1. Vulnerabilidad de las mismas cuentas.

2. Vulnerabilidad de los servicios requeridos para soportar las cuentas.

3. Reducida estabilidad y confiabilidad de la máquina.

4. Alteración inadvertida de la seguridad del host bastión por los usuarios.

5. Incremento en la dificultad para detectar ataques.

Las cuentas de usuario ofrecen avenidas de ataque relativamente sencillas para alguien que desea forzar la entrada del host bastión. Cada cuenta generalmente tiene una contraseña reutilizable que puede ser atacada a través de múltiples medios, incluyendo búsquedas en el diccionario, búsquedas por fuerza bruta o captura al escuchar mientras se está oculto en la red. Multiplique esto por varios usuarios y tendrá un desastre en potencia.

Soportar cuentas de usuario de un modo útil requiere que el host bastión habilite servicios (por ejemplo, servicios de impresión y envío de correo local) que de otra forma podrían ser deshabilitados en él. Cada servicio disponible en el host bastión ofrece otra avenida de ataque a través de problemas en el software o errores de configuración.

Tener que soportar cuentas de usuario también puede reducir la estabilidad y confiabilidad de la misma máquina. Las máquinas que no soportan cuentas de usuario se tienden a operar predeciblemente y son estables. Muchas sitios han encontrado que las máquinas sin usuarios operan mucho más indefiniblemente (o al menos hasta que falla la energía) sin detener el sistema.

Casi siempre es más fácil saber cuando todo está "operando normalmente" en una máquina que no tiene las cuentas de usuario enlodando las aguas. Los usuarios se comportan de modo predecible, pero usted quiere que el host bastión tenga un patrón predecible para detectar intrusos al observar modificaciones en ese patrón.

Si necesita mantener cuentas en el host bastión, manténgalos al mínimo. Agregue cuentas individualmente, revíselas con cuidado y verifique con regularidad que aún sean necesarios.

3.1.8 CÓMO CONSTRUIR UN HOST BASTIÓN.

Ahora se procede a construir un host bastión. Siga estos pasos:

1. Asegure la máquina.
2. Deshabilite todos los servicios no requeridos.
3. Instale o modifique los servicios que quieren proporcionar.
4. Reconfigure la máquina de modo apropiado para desarrollarla hasta su estado final de operación.
5. Haga revisión de seguridad para establecer su "normal operación".
6. Conecte la máquina a la red en la que será usada.

Debe asegurarse con sumo cuidado de que la máquina no sea accesible desde Internet hasta el último paso. Si su sitio todavía no está conectado a esa red, simplemente puede evitar encender la conexión de Internet hasta que el host esté del todo configurado. Si agrega un Firewall a un sitio que ya está conectado a Internet, de configurar el host bastión como una máquina independiente, no conectada a su red.

Aseguramiento de la máquina.

Se debe construir una máquina con un sistema operativo estándar, asegurado lo más posible. Inicie con un sistema virgen.

Comience con una instalación mínima de un sistema operativo virgen.

Comience con una instalación mínima de un sistema operativo virgen, directo de fábrica. Si lo hace así, sabrá exactamente con qué está trabajando. No necesita retroalimentar algo que puede ya tener problemas. Usar tal sistema también facilitara el trabajo posterior.

Cuando configure el sistema operativo, instale lo menos posible. Es mucho más sencillo instalar elementos que borrarlos completamente después.

Arregle todos los problemas conocidos del sistema.

Saque una lista de parches y recomendaciones de seguridad conocidos para su sistema operativo; trabaje sobre ellos para determinar cuales son los relevantes para su propio sistema y corrija todos los problemas descritos en los parches y recomendaciones.

Use una lista de verificación.

Para no pasar nada por alto al asegurar su host bastión, use una lista de verificación de seguridad. Ya existen varias listas de verificación excelentes. Asegúrese de usar la que corresponda a su plataforma y versión de sistema operativo.

Hay listas de verificación más específicas disponibles para sistemas operativos específicos a través de canales de soporte formales o informales para esas plataformas.

Salvaguarda la contabilidad del sistema.

Como un host de seguridad crítica, el host bastión debe almacenar la contabilidad de su sistema. El siguiente paso para construir el host bastión es asegurar que tiene un modo de salvaguardar la contabilidad para el host bastión. La contabilidad del host bastión es importante por dos razones:

1. Son uno de los mejores métodos para determinar si su host bastión está desempeñándose como debe. Si todo lo que hace es almacenado (y debe hacerlo), puede examinarlo para determinar exactamente lo que está haciendo y decidir si es lo que se supone que debe hacer.

2. Cuando algún día alguien tiene éxito en penetrar al host bastión, la información de contabilidad de su sistema es uno de los mecanismos primarios que determinan exactamente qué pasó. Al examinarla y descubrir qué salió mal, debe poder evitar que dicha penetración ocurra de nuevo.

Información de contabilidad del sistema por conveniencia.

La primera copia de la contabilidad del sistema es la que usará de modo regular para monitorear la actividad normal de la máquina. Esta es la información contra las que usted compara sus informes de análisis diarios y semanales.

Puede obtener esta información ya sea del propio host bastión o en algún host interno.

La ventaja de tenerla en el host bastión es la simplicidad: no tiene que darse de alta para ir a otro sistema, ni tiene que configurar los filtros de acceso para hacerlo posible. La ventaja de tenerla en un host interno es la facilidad de acceso.

Información de contabilidad del sistema de catástrofes.

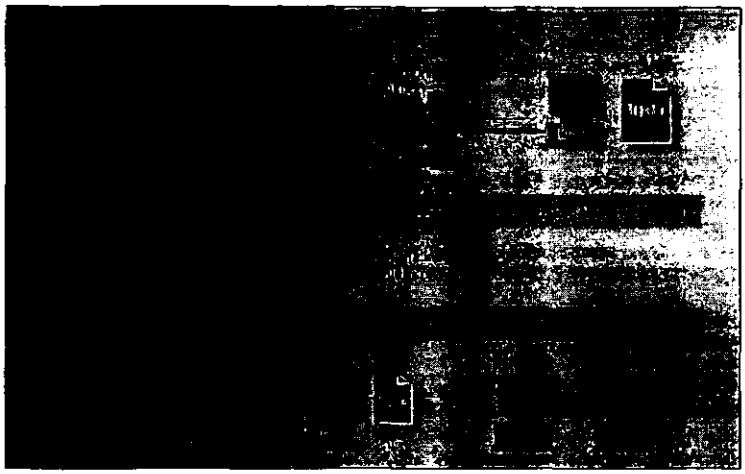
La segunda copia de la contabilidad del sistema es la que usará después de una catástrofe. No puede usar en un momento semejante a la información de contabilidad por conveniencia. Puede ser que no esté disponible, o que ya no esté seguro de su integridad.

Una de las formas más simples de crear información de contabilidad del sistema para catástrofes es conectar una impresora en línea a uno de los puertos seriales del host bastión y simplemente registrar una copia de todo lo que sucede en ese puerto. Sin embargo, se presentan algunos problemas con este procedimiento. Primero, siempre debe mantener la impresora con papel, sin atascos y con una cinta nueva. Segundo, una vez que los registros de contabilidad estén impresos, no hay mucho que pueda hacer con ellos, excepto verlos. Como están de forma electrónica, no hay modo de buscar o analizarlos de forma automatizada.

Cómo configurar el sistema de contabilidad

En un sistema UNIX, el sistema de contabilidad se maneja a través de syslog. Este daemon almacena los mensajes registrados de varios clientes remotos y locales (programas con mensajes que quiere registrar). Cada mensaje se marca con códigos del sitio y de prioridad: los códigos del sitio le dicen a syslog de qué subsistema general es el mensaje (por ejemplo, el sistema de distribución del correo, el kernel, el sistema de impresión, etc.), y los códigos de prioridad le dicen a syslog qué tan importante es el mensaje (puede ser de información de depuración y mensajes informativos de rutina a través de varios niveles, hasta información de urgencia). El archivo /etc/syslog.conf controla lo que hace syslog con los mensajes, basado en su sitio y su prioridad. Un mensaje puede ser ignorado, registrado en uno o más archivos, enviado al daemon syslog en otro sistema, exhibido en las pantallas de ciertos o de todos los usuarios que actualmente utilizan el sistema, o cualquier combinación.

Cuando configure syslog para grabar mensajes en los archivos, puede configurarlo para enviar todos los mensajes a un sólo archivo, o dividir los mensajes entre múltiples archivos con códigos del sitio y prioridad. Si divide los mensajes por códigos del sitio y prioridad, cada archivo será más coherente, pero tendrá que monitorear múltiples archivos: puede ser más fácil encontrar los mensajes de un servicio específico. Por otro lado, si dirige todo a un sólo archivo, sólo tendrá un archivo para buscar todos los mensajes, pero será mucho más grande.



Esté consciente de que los registros remotos vía syslog (por ejemplo, de un router a su host bastión, o de su host bastión a otro host interno) no son cien por ciento confiables. Por un lado, syslog es un servicio basado en UDP, y la persona que envía un paquete UDP no tiene forma de saber si el destinatario lo recibió o no, a menos que éste se lo comunique (el daemon syslog no acusa recibo del remitente). Por otro lado, aun si syslog estuviera basado en TCP, no podría depender absolutamente de que él no perdiera mensajes; ¿que tal si el sistema receptor estuviera fuera de la operación o de algún modo no disponible? Esta es una razón por la que es importante tener una máquina segura conectada localmente para capturar con confianza todos los mensajes de syslog.

A pesar de sus debilidades, syslog es un servicio útil; es recomendable usarlo mucho.

Desactivación de servicios no necesarios.

Una vez que ha completado el proceso básico de asegurar su host bastión, vaya al paso siguiente: deshabilite cualquier servicio que no sea absolutamente necesario y que proporcione el host bastión.

Cualquier servicio provisto por el host bastión podría tener problemas de configuración que podría acarrear problemas de seguridad. Obviamente, tendrá que proporcionar algunos servicios que los usuarios necesitan, siempre que la política de seguridad de su sitio los permita. Pero si el servicio no es absolutamente necesario, no se busque problemas al proveerlo. Si un servicio no lo proporciona el host bastión, no tendrá que preocuparse por posibles problemas de configuración.

Cómo se manejan los servicios

En máquinas UNIX, la mayoría de los servicios se manejan en una de dos formas:

1. Controlando cuándo los inician y quién puede usarlos.

2. Con archivos de configuración de servicios específicos.

Hay dos formas en que se inician los servicios en sistemas UNIX:

1. Al momento de arranque de los archivos /etc/rc de una máquina.

2. A petición del daemon inetd (que a su vez, es iniciado el arranque).

Pocos servicios - por ejemplo, Sendmail - pueden configurarse para ejecutarse bajo alguno de estos mecanismos o por ambos.

Servicios iniciados por archivos /etc/rc.

Los servicios de la primera categoría están diseñados para ejecutarse indefinidamente. Son iniciados una vez (cuando la máquina arranca) y nunca se supone que terminan (por supuesto, algunas veces sí terminan, ya sea porque son eliminados por un administrador de sistema o porque encuentran un problema u otro error). Los servidores están configurados de esta manera si necesitan manejar pequeñas transacciones rápidamente, o si necesitan "recordar" información. Configurarlos así evita los retrasos asociados con empezar una nueva copia del servidor para manejar cada solicitud hecha a él.

Los servidores de este tipo son iniciados desde los archivos /etc/rc de un sistema UNIX, que son scripts (guiones) de shell ejecutados cuando la máquina arranca. Ejemplos de servidores de este tipo son los que manejan NFS, SMTP y DNS. En versiones de UNIX basadas en BSD, hay generalmente algunos archivos en /etc con nombres que comienzan con "rc" (por ejemplo, /etc/rc.boot). En versiones de UNIX basadas en System V, generalmente hay directorios en /etc en vez de archivos (por ejemplo, /etc/rc.0.d); los directorios contienen los diversos comandos de inicio, cada uno en su propio archivo pequeño.

En cualquier caso, debe ser cuidadoso al observar todos los guiones de inicio y todos los que ellos llaman en forma recursiva. Por lo común, más de un guión será ejecutado en el proceso de arranque de sistema completo. En sistemas UNIX modernos, esos guiones con frecuencia a otros, algunas veces a través de múltiples niveles de indirección. Por ejemplo, puede encontrar que un guión de inicio llama a otro guión para iniciar los servicios de red, y ese llama a otro más para iniciar el servicio de archivos. También puede encontrar que los guiones de inicio utilizan opciones místicas para comandos familiares (por ejemplo, generalmente ejecutan ifconfig con opciones poco usadas que ocasionan que ifconfig recoja información de configuración de lugares oscuros).

Servicios iniciados por inetd.

Algunos servidores están diseñados para iniciar a "petición", y para salir después de proporcionar el servicio solicitado. Tales servidores se usan generalmente para servicios que se solicitan con poca frecuencia; para servicios que no son sensibles a retrasos por iniciar un nuevo servidor desde cero, y para servicios que necesitan un nuevo proceso servidor a fin de hacerse cargo de la solicitud (por ejemplo, sesiones de Telnet o FTP, donde se usa un servidor independiente para cada sesión activa).

Los servidores de este tipo casi siempre se ejecutan desde el servidor inetd (el mismo servidor inetd, porque es ejecutado indefinidamente, se inicia desde los archivos /etc/rc). El servidor inetd atiende solicitudes de servicios especificados en el archivo de configuración /etc/inetd.conf. Cuando escucha tal solicitud, inicia el servicio adecuado para procesarla.

Que servicios dejar habilitados

Ciertos servicios son esenciales para la operación de la máquina y tal vez deba dejarlos habilitados, sin importar para qué nada más está configurada la máquina. En un sistema UNIX, estos procesos incluyen:

init, swap y page

Los tres seudoprocesos de kernel usados para manejar todos los demás procesos.

cron
Ejecuta otros trabajos en tiempos fijos, para mantenimiento y otras tareas.

syslogd
Colecta y graba mensajes registrados del kernel y otros programas.

inetd
Arranca los servidores de la red (como Telnetd y FTPd) cuando tales servicios son solicitados por otras máquinas.

Además obviamente necesitará los procesos del servidor para los servicios que ha decidido proporcionar en su host bastión, por ejemplo, servidores reales o proxy para Telnet, FTP, SMTP, y DNS.

Qué servicios debe deshabilitar

Querrá deshabiltar todos los servicios excepto los que ha decidido proporcionar, y los de soporte necesarios para que éstos se ejecuten. Quizá no siempre sepa cuáles son los servicios de soporte requeridos, en especial porque los nombres en UNIX tienden a ser poco informativos.

¿Cómo saber cuáles servicios deshabilitar?

Hay tres reglas simples que debe aplicar:

1. Si no lo necesita, desactívelo.
2. Si no sabe qué hace, desactívelo.
3. Si desactivarlo causa problemas, ahora sabe lo que hace y puede activarlo de nuevo (si en verdad es necesario) o ver qué hacer sin él.

NFS y servicios relacionados.

Comience con NFS y servicios de red relacionados. No va a necesitarlos. Ninguna máquina interna debe confiar en su host bastión lo suficiente para dejarlo montar los discos internos de la máquina mediante NFS.

Además, quizá no habrá nada en el host bastión que quiera exportar vía NFS, el cual es muy útil pero muy poco seguro.

Los servicios NFS son provistos por un grupo de servidores; el grupo específico de éstos y los nombres de los servidores individuales varían ligeramente de una versión de UNIX a otra. Por ejemplo:

* nfsd
* biod
* mountd
* statd
* lockd
* automount
* keyserv
*rquotad
*amd

Muchos de estos servicios se ejecutan en el momento del arranque desde los archivos /etc/rc, aunque algunos son iniciados a petición por inetd, mountd es algo peculiar, pues con frecuencia se ejecuta al momento del arranque y está listado en el archivo de configuración de inetd, aparentemente de modo que pueda reiniciarse si por alguna razón la copia iniciada al momento del arranque es detenida.

Otros servicios RPC

También debe deshabilitar otros servicios basados en el sistema RPC. El más importante de ellos es NIS/YP, un servicio proporcionado por los siguientes servidores:

* ypserv
*ypbind
*ypupdated

Estos servidores arrancan por lo general al momento del arranque de los archivos /etc/rc.

También deshabilite estos servicios basados en RPC:

* rexd (el servicio de ejecución remota, iniciado por inetd)
* wald ("escribir a todos", o daemon wall, iniciado por inetd)

Servicios de arranque

Su host bastión tal vez no debe proporcionar servicios de arranque a otras máquinas, nada debe confiar los suficiente en el host para querer arrancar de él, lo cual significa que, en muchos casos, debe deshabilitar estos servicios:

* tFTPd
* bootd
* bootpd

Servicios BSD de comando 'r'

Debe deshabilitar todos. Los servidores para estos servicios se llaman, por lo común, rshd, rlogin y rexecd y casi siempre los arranca inetd. Los servicios 'r' restantes basados en éstos no se ejecutarán sin ellos.

Enrutamiento.

Otro servidor que su host bastión quizá no necesite es routed. Este servidor entrará en funcionamiento en el momento del arranque inicial de los archivos /etc/rc, atiende transmisiones de información de enrutamiento y actualiza la tabla de enrutamiento del kernel basado en lo que escucha.

Probablemente, quizá no necesite routed en su host bastión, porque éste último quizá esté localizado en el perímetro de su red, donde el enrutamiento debe ser muy sencillo. Un camino más seguro es crear rutas estáticas que apunten hacia sus redes internas y una ruta predeterminada que apunte hacia su router de compuerta a Internet. Esto se hace al momento del arranque añadiendo comandos route a los archivos /etc/rc.

Si desea hacer un enrutamiento dinámico en el host bastión, obtenga y use gated en lugar de routed, gated entiende los mimos protocolos de enrutamiento que routed (y varios más).

FTPd

Si va a proporcionar servicio FTP anónimo en su host bastión, debe configurar apropiadamente el servidor FTP. Debe reemplazar el programa FTPd con uno o más apropiado para brindar el servicio FTP anónimo que los programas estándar FTPd que entregan muchos proveedores de UNIX.

Si no va a brindar FTP anónimo, puede deshabilitar su servidor FTP por completo, inetd lo inicia a petición.

Desactive el enrutamiento.

Si tiene un host con doble acceso que no se supone sea un router, debe deshabilitar el enrutamiento. Para actuar como un router IP, un host con doble acceso debe aceptar paquetes dirigidos a direcciones IP de otras máquinas y enviarlos a ellas correctamente. Esto se conoce como IP forwarding (direccionamiento IP) y usualmente se implementa a bajo nivel en el kernel del sistema operativo

Montaje de archivos de sistema como de sólo lectura.

Una vez que tiene configurado el host bastión, no quiere que nadie (en especial un atacante) pueda cambiar la configuración. Para evitar que esto pase, monte los archivos del sistema del host bastión como de sólo lectura si es posible (sobre todo los archivos de sistema que contienen programas binarios) para protegerlos de obstrucciones.

Es mucho mejor si puede usar hardware protegido contra escritura; un atacante puede montar discos con el permiso de escritura sin tener acceso físico a la máquina, pero no va a servir de nada si el hardware para proteger contra escritura está activado en el disco.

Con muchas versiones de UNIX, también tendrá que ofrecer espacio de disco para escribir el intercambio de memoria o desactivar el intercambio (swap). Muchas versiones de UNIX no permiten que desactive el intercambio; sin embargo, por lo general le permitirán usar un disco independiente para espacio de intercambio, y ese disco puede estar disponible para escritura con seguridad.

Ejecución de una auditoría de seguridad.

Una vez configurado el host bastión, el siguiente paso es ejecutar una auditoría de seguridad. Hay dos razones para hacerlo. Primero, es una forma de asegurarse de que no ha omitido nada durante la instalación del sistema. Segundo, establece una "línea base", o una base de comparación, contra la que puede comparar auditorías futuras. Así podrá detectar cualquier descompostura en la máquina.

Paquetes de auditoría

Muchos paquetes de auditoría tienen dos propósitos básicos:

* Verificar la existencia de fallas conocidas de seguridad. Estas fallas han sido descubiertas por administradores de sistemas y explotadas por los atacantes en las entradas no deseadas al sistema, o documentadas en los libros y papeles de seguridad en las computadoras.

* Establecer una base de datos de sumas de verificación de todos los archivos en un sistema; hacerlo le permite al administrador del sistema reconocer cambios futuros en los archivos, particularmente cambios no autorizados.

Use los paquetes de auditoría

¿Cómo usar los diversos paquetes de auditoría para revisar su sistema? Los detalles de lo que debe hacer dependen de qué paquete use.

Debe hacer algunas configuraciones. No sólo instale el programa, ejecútelo y espere obtener resultados favorables. Prepárese a ejecutar al paquete de auditoría varias ocasiones, recibir advertencias y reconfigurar la máquina o el paquete de auditoría para deshacerse de las advertencias.

Una vez que ha usado las herramientas descritas para crear su línea base inicial, almacene una copia de las herramientas y los resultados de esta revisión inicial en un lugar seguro.

Periódicamente (a diario o una vez a la semana, por decir algo, dependiendo de las necesidades y capacidades de su propio sitio, revise la máquina una vez más y compare la nueva revisión con la línea base. Asegúrese de que puede explicar cualquier diferencia que encuentre. Lo ideal es que haga que esta nueva auditoría periódica sea automática, a fin de que ocurra con regularidad y confiabilidad. Desafortunadamente, no es tan sencillo como suena.

Sobre sumas de verificación para auditoría.

Las sumas de verificación son muy útiles para las auditorías. Un intruso que cambia un programa o archivo de configuración corregirá después casi con exactitud las fechas de la modificación para que usted no pueda usarlas como un índice confiable. Comparar cada archivo con una copia de la línea base evita el problema, pero toma mucho tiempo y requiere que almacene una copia de cada archivo, duplicando así sus requisitos de almacenamiento. Las sumas de verificación son, probablemente, la mejor opción.

Una suma de verificación es un número calculado del contenido del archivo que cambiará si el archivo cambia. El cálculo de la suma de verificación lleva tiempo, pero no tanto como leer todo dos veces para hacer una comparación bit por bit. Además, almacenar las sumas de verificación ocupa mucho menos espacio que guardar el archivo completo. Sin embargo, las sumas de verificación no son representaciones completas del archivo y cada algoritmo de la suma tiene casos en qué dará la misma suma de verificación, es menos probable que los archivos con la misma suma se parezcan el uno al otro en alguna forma.

3.1.9 CÓMO OPERAR EL HOST BASTIÓN.

Una vez que ha puesto a funcionar el host bastión, su trabajo acaba de comenzar. Debe vigilar muy de cerca sus operaciones.

Aprenda qué es el perfil de uso normal.

Si va a monitorear el host bastión buscando anormalidades que podrían indicar intrusiones u otros tipos de compromisos del sistema, primero debe desarrollar un conocimiento de cuál es el perfil de uso "normal" en el host bastión. Haga preguntas como las siguientes, o similares:

* ¿Cuántos trabajos tienden a ser ejecutados a la vez?
* ¿Relativamente cuánto tiempo de CPU consumen cada uno estos trabajos?
* ¿Cuál es la carga típica en diferentes horas del día?

Su meta es desarrollar un conocimiento casi intuitivo de cómo se ejecuta su sistema normalmente, para que con rapidez pueda reconocer e investigar una actividad anormal.

Considere la opción de escribir software para monitoreo automático.

Es difícil hacer un trabajo de monitoreo del sistema a fondo. Aunque los registros que produce su sistema proporcionan mucha información útil, es fácil que lo abrume por el diáfano volumen de datos registrados. La información importante con frecuencia puede estar oculta. Con suma frecuencia, los registros acaban por usarse sólo después de una intrusión, cuando, de hecho, podrían emplearse para detectar (y, por lo tanto, quizá detener) una intrusión mientras está ocurriendo.

Como cada sistema operativo y cada sitio son distintos, cada host bastión se configura diferente y cada sitio tiene ideas distintas sobre cuál debe ser la respuesta de un sistema de monitoreo. Por ejemplo, algunos quieren correo electrónico; algunos quieren que la salida de información sea alimentada a un sistema de manejo existente basado en SNMP, algunos quieren que los sistemas envíen un mensaje a los paginadores de los administradores de sistemas.

Como proteger la máquina y las copias de respaldo.

Una vez que el host bastión está completamente configurado y en operación, proteja la máquina física y asegúrese que sus copias de respaldo están protegidos del robo u otro compromiso.

Vigile cuidadosamente los rearranques.

¿Cómo sabrá si alguien ha burlado la seguridad? A veces es bastante obvio. Pero en ocasiones, tendrá que sacar conclusiones del comportamiento del sistema. Rearranques inexplicables o tiempo apagado del sistema pueden ser la clave. Muchos ataques (modificar un kernel por ejemplo), no pueden tener éxito a menos que el sistema sea rearrancado.

En el host bastión, las trabas del sistema y rearranques casi no deben ocurrir. Una vez que el host bastión esté completamente configurado y en operación, debe ser estable, ejecutado con frecuencia por semanas o meses de libertad sin una traba del sistema o rearranque. Si ocurre una detención del sistema o rearranque, investíguelo de inmediato para determinar si fue causado por un problema legítimo o puede ser un resultado de alguna clase de ataque.

Haga copias de respaldo seguras.

Las copias de respaldo en un host bastión son engañosas por razones de confianza. ¿En quién puede confiar?

Definitivamente, no quiere que las máquinas internas confíen demasiado en el host bastión para que éste haga copias de respaldo en sus unidades de cinta. Si el host bastión ha sido comprometido de algún modo, puede ser desastroso. Tampoco quiere que el host bastión confíe en las máquinas internas, esto podría derivar en la alteración del host bastión por usuarios internos (bien intencionados), o en un ataque de algún host fingiendo ser un sistema interno.

Por lo tanto debe hacer, por lo general, copias de respaldo a un mecanismo de cinta conectado directamente al host bastión. Bajo ninguna circunstancia debe confiar en respaldar el host bastión en discos que permanezcan conectados a él. Debe hacer copias de respaldo que sean removidas del host bastión para que no permitan el acceso a un atacante que lo comprometa.

Como con todos los respaldos de todos los sistemas, debe vigilar los respaldos de todos los sistemas, debe vigilar los respaldos de su host bastión como si vigilara la máquina misma, pues contienen toda la información de configuración. Un atacante que tenga acceso a estas copias de respaldo será capaz de analizar la seguridad de su host bastión sin siquiera tocarlo. Con la información que proporcionan estos respaldos, posiblemente encontrará un modo de penetrar al sistema sin activar ninguna de las alarmas del host bastión.

3.2 FILTRADO DE PAQUETES

El filtrado de paquetes es un mecanismo de seguridad de la red que funciona controlando qué información puede fluir de y hacia una red.

Para transferir información a través de una red de dividirse en pequeñas partes, cada una de las cuales se envía en forma separada. Dividir la información en partes permite que varios sistemas compartan la red: cada uno envía fragmentos por turnos. En una red IP, esos pequeños elementos de información se llaman paquetes. Toda la información transferida a través de las redes IP se hace en forma de paquetes.

El mecanismo básico que interconecta las redes IP se llama router. Un router puede ser una pieza dedicada de hardware sin algún otro propósito, o puede ser una pieza de software que se ejecuta en un sistema UNIX o en una computadora personal (con MS-DOS, Windows, Sistema de Mcintosh u otro) para uso general. Los paquetes que atraviesan una red de Internet viajan de un router a otro hasta que llegan a su destino. La propia Internet es algo así como el abuelito de las intranets: es la máxima "red de redes".

Un router tiene que tomar una decisión acerca del enrutamiento de cada paquete que recibe; debe decidir cómo enviar ese paquete hasta su destino final. En general, para ayudar al router en su decisión, un paquete no lleva más información que la dirección IP de su destino final. El paquete le dice al router a dónde quiere ir, pero no cómo llegar ahí. Los routers se comunican entre sí usando "protocolos de enrutamiento", como Routing Information Protocol (RIP) y Open Shortest Path First (OSPF) para construir las tablas de enrutamiento en la memoria del router a fin de determinar cómo llevar los paquetes a su destino. Cuando enruta un paquete, el router compara la dirección destino que tiene aquél con los registros en la tabla de enrutamiento y lo envía tal como indica la tabla. Con frecuencia, no hay una ruta específica para un destino específico, y el router usa una "ruta predeterminada"; en general, tal ruta dirige el paquete hacia routers más inteligentes o mejor conectados (en la mayoría de los sitios, las rutas predeterminadas apuntan hacia Internet).

Para determinar cómo enviar un paquete a su destino, un router común ve únicamente la dirección destino del paquete y sólo pregunta ¿Cómo puede enviarlo? Un router con filtrado de paquetes también considera esta pregunta ¿Debo enviar este paquete? El router con filtrado de paquetes responde de acuerdo con la política de seguridad que se programó por medio de las reglas para el filtrado de paquetes.

3.2.1 POR QUÉ FILTRADO DE PAQUETES.

El filtrado de paquetes le permite controlar (permite o niega) la transferencia de información con base en:

* La dirección de donde (supuestamente) proviene la información.
* La dirección adonde se dirige la información.
* Los protocolos de nivel sesión y aplicación que se emplean para transferir la información.

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

No permita que alguien use Telnet (protocolo de aplicación) para permitir acceso l sistema desde el exterior

ó

Permita a todos enviar correo electrónico por medio de SMTP (protocolo de aplicación)

o incluso:

Esa máquina puede enviarnos noticias por medio de NNTP (protocolo de aplicación), pero ninguna otra puede hacerlo.

Sin embargo, no le permitirá decir:

Este usuario puede utilizar Telnet desde afuera, pero ningún otro puede hacerlo.

Esto es así porque "usuario" no es algo que el sistema de filtrado de paquetes pueda identificar. Y no le permite decir:

Puede transferir estos archivos pero no estos otros.

Esto es porque "archivo" tampoco es algo que los sistemas de filtrado de paquetes puedan identificar.

La ventaja principal del filtrado de paquetes es concentrar: le permite proporcionar, en un sólo lugar, protecciones específicas para toda una red. Considere el servicio de Telnet como un ejemplo. Si lo deshabilita apagando el servidor de Telnet en todos sus hosts, aún debe preocuparse porque alguien en su organización instale una máquina (o reinstale una vieja) con el servidor de Telnet. Por otra parte, si Telnet no es aceptado por su router, tal máquina estará protegida desde el principio, sin importar si el servidor Telnet estaba o no ejecutándose en ese momento.

Los routers también presentan un punto de choque útil para todo el tráfico que entra o sale de una red. Incluso si tiene múltiples routers por redundancia, probablemente tendrá muchos menos, bajo un control más estricto, que máquinas anfitriones.

Ciertas protecciones pueden proporcionarse sólo con los routers de filtrado, y sólo cuando se colocan en lugares específicos de la red. Por ejemplo, es buena idea regresar todos los paquetes que tiene direcciones fuente internas (en otras palabras, paquetes que dicen venir de máquinas internas pero que verdaderamente vienen del exterior) porque son, por lo general, parte de los ataques de falsificación de dirección. En tales casos, un atacante finge venir de una máquina interna. Una toma de decisión de esta clase puede hacerse sólo por con router en el perímetro de la red. Sólo un router en esa posición (que es, por definición, la frontera entre "dentro" y "fuera") será capaz de reconocer tal paquete, al observar la dirección fuente y si el paquete vino desde dentro (la conexión interna de la red) o desde fuera (la conexión externa de la red).

3.2.2 VENTAJAS DEL FILTRADO DE PAQUETES.

El filtrado de paquetes tiene una serie de ventajas.

Un router de protección puede ayudar a proteger toda una red.

Una de las ventajas clave del filtrado de paquetes es que un solo router con filtrado de paquetes colocado estratégicamente puede ayudar a proteger toda una red. Si sólo hay un router que conecta su sitio a Internet, logra una enorme ventaja en la seguridad de la red, sin importar el tamaño de su sitio, al hacer el filtrado de paquetes en ese router.

El filtrado de paquetes no necesita conocimiento o cooperación del usuario.

A diferencia de un proxy, el filtrado de paquetes no requiere ningún programa o configuración especial de las máquinas cliente, ni necesita algún adiestramiento especial o procedimiento por parte de los usuarios. Cuando un router con filtrado de paquetes decide dejar pasar un paquete el router es idéntico a un router común. Lo ideal es que los usuarios ni siquiera se percaten de que está presente, amenos que intenten hacer algo prohibido por la política de filtrado del router.

Esta "transparencia" quiere decir que el filtrado de paquetes puede hacerse sin la cooperación y con frecuencia sin el conocimiento de los usuarios. La cuestión no es que usted puede hacer esto de modo subversivo, a espaldas de sus usuarios (aunque acciones como la anterior a veces son necesarias - todo depende de las circunstancias -, pueden ser sumamente políticas). Lo importante es que puede hacer filtrado de paquetes sin que ellos tengan que aprender algo nuevo para que funciones y sin que usted tenga que depender de que ellos hagan (o no) algo para que funcione.

3.2.3 DESVENTAJAS DEL FILTRADO DE PAQUETES.

Aunque el filtrado de paquetes tiene muchas ventajas, tiene también algunas desventajas.

Las herramientas actuales para filtrado no son perfectas.

A pesar de la amplia disponibilidad para filtrado de paquetes en varios productos de hardware y software, el filtrado aún no es una arma perfecta. En mayor o menor grado, la capacidad de filtrado de paquetes en muchos de estos productos comparten limitaciones comunes:

- Las reglas para filtrado tienden a ser difíciles de configurar. Aunque hay un rango de dificultad, habitualmente va desde lo difícil, con un ligero dolor de cabeza, hasta lo imposible, con una fuerte jaqueca.

- Una vez configuradas, las reglas para el filtrado de paquetes tienden a ser difíciles de probar.

- Las capacidades para filtrado de paquetes en muchos de los productos están incompletas, lo cual hace difícil o imposible la implementación de ciertos tipos de filtros sumamente deseables.

- Como cualquier cosa, los programas para filtrado de paquetes pueden tener problemas, los cuales son más propensos a convertirse en problemas de seguridad que los errores proxy. Por lo general, un proxy que falla simplemente deja pasar información, mientras una falla de filtrado de paquetes podría permitir la entrada de paquetes que debieron rechazarse.

Algunos protocolos no están diseñados correctamente para el filtrado de paquetes.

Aun con usos perfectos de filtrado de paquetes, encontrará, que algunos protocolos no están correctamente diseñados para dar seguridad por medio del filtrado de paquetes. Tales protocolos incluyen los comandos "r" y protocolos basados en RPC, como NFS y NIS/YP.

Algunas políticas no pueden aplicarse de inmediato por medio de routers comunes con filtrado de paquetes.

La información de que dispone en router con filtrado de paquetes no permite especificar ciertas reglas que a usted le gustaría tener. Por ejemplo, los paquetes dicen de qué host vienen pero generalmente no de qué usuario. Por eso, no puede aplicar restricciones a usuarios específicos. De modo similar, los paquetes muestran hacia qué puerto van pero no hacia qué aplicación; cuando aplica restricciones a los protocolos de nivel más alto, lo hace por número de puerto con la esperanza de que ningún otro programa esté ejecutándose en el puerto asignado a ese protocolo. Intrusos maliciosos podrían alterar fácilmente esta clase de control.

3.2.4 CÓMO CONFIGURAR UN ROUTER CON FILTRADO DE PAQUETES

Para configurar un router con filtrado de paquetes, primero debe decidir qué servicios quiere permitir o llegar, y después convertir sus decisiones en reglas sobre los paquetes. En realidad, quizá no le importen los detalles de los paquetes en sí, lo que quiere es que su trabajo se haga. Por ejemplo, quiere recibir correo de Internet, y si eso lo manejan los paquetes o el fantasmas de Murphy le es irrelevante. Por otro lado, al router, sólo le importan los paquetes; más aún, sólo una parte muy limitadas de ellos. Para construir las reglas de sus routers, debe traducir el enunciado general "Recibir correo de Internet" a una descripción de las clases específicas de paquetes que desea que el router deje pasar.

Usualmente los protocolos son bidirediccionales.

Por lo general, los protocolos son bidireccionales; casi siempre incluyen un extremo que envía una solicitud o comando, y otro que envia una respuesta de alguna especie. Cuando diseñe las reglas para el filtrado de paquetes, debe recordar que los paquetes viajan en ambas direcciones. Por ejemplo, no sirve de nada permitir paquetes Telnet de salida que enviarán sus golpes de teclado a un host remoto si tampoco permite que regresen los paquetes para esa conexión que traen el despliegue en pantalla.

Por el contrario, tampoco sirve de nada bloquear media conexión. Muchos ataques pueden llevarse a cabo si los atacantes pueden introducir paquetes a su red, aun si no pueden obtener respuestas. Esto es posible porque las respuestas pueden ser lo suficientemente predecibles para permitir que los atacantes lleven a cabo su parte de la conversación si tener que ver las respuestas. Si éstas son predecibles, un atacante no necesita verlas. No podrá extraer ninguna información directamente si no las ve, pero puede hacer algo que le información indirectamente. Por ejemplo, aunque el atacante no pueda ver directamente el archivo /etc/passwd, quizá puedan ejecutar un comando para que le envíe una copia.

Permiso predeterminado contra negativa predeterminada.

Desde el punto de vista de seguridad, es mucho más seguro tomar ésta actitud: las cosas pueden negarse de forma predeterminada. Sus reglas para el filtrado de paquetes deben reflejar esta postura. Comience con una posición de negar todo y después establezca reglas que permitan sólo los protocolos necesarios, de los que entienden las implicaciones para la seguridad y siente que puede funcionarle con la suficiente seguridad (de acuerdo con su definición particular de "suficiente seguridad).

La postura de negativa predeterminada es más segura y efectiva que la de permiso predeterminado, el cual incluye permitir todo en forma predeterminada y después tratar de bloquear aquello que sabe que ocasiona problemas. La realidad es que usted nunca conocerá todos los problemas y, así, nunca será capaz de hacer trabajo completo.

En términos prácticos, la postura de negativa predeterminada significa que sus reglas para el filtrado deben ser una pequeña lista de tareas específicas que sí permite, tal vez con ciertas cosas muy específicas que niega y que están esparcidas para que la lógica dé buenos resultados, seguidas por una negativa predeterminada que abarque todo lo demás.

3.2.5 CÓMO ES UN PAQUETE

Para entender el filtrado de paquetes, primero debe entender los paquetes y cómo se manejan en cada nivel del conjunto de protocolos TCP/IP:

- Nivel de aplicación (por ejemplo, FTP, Telnet, HTTP)
- Nivel de transporte (TCP o UDP)
- Nivel de Internet (IP)
- Nivel de acceso a la red (Por ejemplo, Ethernet, FDDI, ATM)

Los paquetes se construyen de tal modo que los niveles de cada protocolo usados para una conexión específica los envuelven, como las capas de una cebolla.

En cada nivel, un paquete tiene dos partes: el encabezado y el cuerpo. El encabezado contiene información del protocolo relevante para ese nivel; el cuerpo, la información de ese nivel que con frecuencia consiste en un paquete completo para el siguiente nivel en el conjunto. Cada nivel trata como datos la información que obtiene del nivel superior y les aplica su propio encabezado.

En cada nivel, el paquete contiene toda la información enviada del nivel superior; nada se pierde. Este proceso de preservar la información mientras se coloca un nuevo encabezado se conoce como encapsulamiento.

En el nivel de aplicación, el paquete consiste simplemente en la información que será transferida (por ejemplo, parte de un archivo que va a transferirse durante una sesión FTP). Mientras viaja al nivel de transporte, el Protocolo de Control de Transmisión o TCP, o el Protocolo de Datagrama de Usuario o UDP conservará la información del nivel anterior y le colocará un encabezado. En el siguiente nivel, IP considera el paquete completo (que ahora consta del encabezado TCP o UDP más la información) como información y le coloca su propio encabezado IP. Por último, en el nivel de acceso a la red, Ethernet u otro protocolo de red considera todo el paquete IP como información y le pone su propio encabezado. La figura siguiente muestra cómo funciona este proceso.

En el otro lado de la conexión el proceso se invierte. Conforme pasa la información de un nivel al siguiente nivel superior, cada encabezado (cada capa de la cebolla) será quitado por su nivel respectivo. Por ejemplo, el nivel de Internet quita el encabezado IP antes de pasar la información encapsulada al nivel de transporte (TCP o UDP).

Ejemplo de TCP/IP/Ethernet

Consideremos un ejemplo de un paquete TCP/IP (por poner un caso, uno que es parte de una conexión Telnet) en una red Ethernet. Hay cuatro niveles que nos interesan aquí: el nivel de Ethernet, el nivel de IP, el nivel de TCP y el nivel de información.

Nivel de Ethernet

En el nivel de Ethernet, el paquete consta de dos partes: el encabezado Ethernet y el cuerpo Ethernet. En general, usted no podrá hacer filtrado de paquetes basado en la información del encabezado Ethernet. Básicamente, el encabezado le dice:

* Qué clase de paquete es este; en este ejemplo, supondremos que es un paquete IP.
* La dirección Ethernet de la máquina que coloca el paquete en este segmento específico de la red Ethernet, o la máquina fuente original, si está conectada a este segmento, de otro modo, el último router en la trayectoria que va de la máquina de origen hasta aquí.
* La dirección Ethernet destino de los paquetes en este segmento específico de la red Ethernet, o tal vez la máquina destino, si está conectada a este segmento; de otro modo, el siguiente router en la trayectoria que va de aquí a la máquina destino.

Como estamos considerando paquetes IP en este ejemplo, sabemos que el cuerpo Ethernet contiene un paquete IP.

Nivel IP

En el nivel IP, el paquete IP se compone de dos partes: el encabezado IP y el cuerpo IP. Desde el punto de vista del filtrado de paquetes, el encabezado IP contiene cuatro elementos de información interesantes:

* La dirección IP fuente: de cuatro bytes de longitud y, por lo general, escrita más o menos como 172.16.244.34.

* La dirección IP destino, igual que la dirección fuente.

* El tipo de protocolo IP, que identifica el cuerpo como un paquete TCP, en contraste con el paquete (datagrama) UDP, un paquete de Protocolo de Control de Mensajes de Internet (ICMP) o algún otro tipo de paquetes.

* El campo de opciones IP, que casi siempre está vacío pero que es donde estarían especificadas ciertas opciones, como la ruta IP fuente y las opciones de seguridad IP si se usaran para un paquete dado.

IP puede dividir un paquete demasiado largo para cruzar en una serie de paquetes menores, llamados fragmentos. Fragmentar un paquete no cambia su estructura en el nivel IP (los encabezados IP se duplican en cada fragmento), pero puede significar que el cuerpo contenga sólo una parte de un paquete en el siguiente nivel.

El cuerpo IP en este ejemplo contiene un paquete TCP no fragmentado, aunque bien podría contener el primer fragmento de un paquete TCP fragmentado.

Nivel TCP

En el nivel TCP, el paquete contiene también dos partes: el encabezado TCP y el cuerpo TCP. .Desde el punto de vista del filtrado de paquetes, el encabezado TCP contiene tres piezas de información valiosas:

- El puerto TCP fuente, con un número de dos bytes que especifica de qué cliente o proceso servidor en la máquina de origen proviene el paquete.

- El puerto TCP destino, igual que el puerto TCP fuente.

- El campo de banderas TCP.

El campo de banderas TCP contiene un bit de interés para el filtrado de paquetes: el bit ACK (de confirmación de recepción). Al examinar este bit ACK, un router con filtrado de paquetes puede determinar si un paquete específico es el que inicia una conexión TCP (si el bit ACK no está encendido) o es un paquete subsecuente (si el bit ACK está encendido). El bit ACK es parte del mecanismo de TCP que garantiza el envío de información. El bit ACK se activa siempre que un extremo de la conexión ha recibido información del otro extremo (confirma la información recibida). Por lo tanto; el bit ACK se activa en todos los paquetes que van en cualquier dirección, excepto el primer paquete del cliente al servidor.

El cuerpo TCP contiene los "datos" reales que van a trasmitirse (por ejemplo, para Telnet los golpes del teclado o los desplegados e pantalla que son parte de una sesión Telnet; para FTP, la información que se está transfiriendo o los comandos ejecutándose como parte de una sesión FTP).

IP

IP sirve como una zona intermedia común para Internet. Puede tener muchos niveles debajo, como Ethernet, Token Ring, FDDI, PPP. Ip puede tener muchos otros protocolos en los niveles superiores, siendo los más comunes TCP, UDP e ICMP, al menos fuera de los ámbitos de investigación.

Opciones IP

Como se vió anteriormente los encabezados IP incluyen un campo de opciones, que por lo general está vacío. En su diseño, se buscó que tal campo fuera un lugar para contener información especial o instrucciones para el manejo del paquete que no tuvieran un campo específico propio en el encabezado. Sin embargo, los diseñadores de TCP/IP hicieron tan buen trabajo al dar campos para todo lo necesario que el campo de opciones casi siempre está vacío. En la práctica, las opciones IP rara vez se utilizan, excepto para intentos de penetración y, muy de vez en cuando, para depurar la red.

La opción IP más común con la que podría enfrentarse un Firewall es la de enrutamiento IP fuente. El enrutamiento fuente permite a la máquina que origina un paquete especificar la ruta que se supone deberá seguir el paquete para llegar a su destino, en lugar de dejar a cada router usar sus tablas de enrutamiento para decidir a dónde enviarlo. Se supone que el enrutamiento fuente debe anular las instrucciones de las tablas de enrutamiento. En teoría, la opción de enrutamiento fuente es útil para anular routers con tablas descompuestas o incorrectas; si sabe la ruta que debe tomar el paquete pero las tablas de enrutamiento están descompuestas, puede anular esa información errónea especificando las opciones apropiadas del enrutamiento IP fuente en todos sus paquetes. Sin embargo, en la práctica, el enrutamiento fuente se usa comúnmente sólo por los atacantes que intentan burlar las medidas de seguridad haciendo que los paquetes sigan caminos inesperados.

Muchos sistemas de filtrado eliminan cualquier parte del paquete que incluya una opción IP, sin por lo menos tratar de descubrir cuál es o qué significa; por lo general, esto parece funcionar bien, sin causar ningún problema especial.

Fragmentación IP

Otra consideración del nivel IP para filtrado de paquetes es la fragmentación. Una de las características de IP es su habilidad para dividir un paquete grande que, de otra forma, no podría atravesar algún enlace en la red (por las limitantes de tamaño de paquetes en ese enlace), en paquetes más pequeños, llamados fragmentos, que pueden cruzar ese enlace. Después, la máquina destino vuelve a unirlos en un paquete completo (no es la máquina del otro lado del enlace limitado,; una vez que un paquete es fragmentado, permanece así hasta llegar a su destino).

Desde el punto de vista del filtrado de paquetes, el problema de la fragmentación es que sólo el primer fragmento contendrá en el encabezado la información para los protocolos del nivel superior, como TCP, que el sistema para filtrado de paquetes necesita para decidir si permite o no todo el paquete. La reacción común del filtrado de paquetes con respecto a la fragmentación es permitir el paso de fragmentos no iniciales, y hacer el filtrado sólo en el primer fragmento de un paquete. Esto es seguro ya que, si el filtrado decide eliminar el primer fragmento, el sistema destino no será capaz de reunir el resto de los fragmentos del paquete original, sin importar cuántos reciba. Si no puede reconstruir el paquete original, el paquete reunido en forma parcial no será aceptado.

El host destino guarda los fragmentos en la memoria durante cierto lapso, esperando obtener la pieza faltante; esto hace posible que los atacantes usen los paquetes fragmentados en un ataque de negación del servicio. Cuando el host destino se da por vencido para unificar el paquete, envía un mensaje de ICMP de "tiempo expirado para unificación de paquete" el host fuente, que le dirá al atacante que existe el host y por qué no tuvo éxito la conexión. No hay nada que hacer con tal ataque de negación de servicio, pero puede filtrar los mensajes de ICMP.

Los fragmentos de salida podrían, de algún modo, contener información que usted no quiere divulgar al mundo externo. Por ejemplo, un paquete NFS de salida sería fragmentado casi con certeza y, si el archivo era confidencial, la información sería divulgada. Si esto ocurre por accidente, es poco probable que sea un problema; por lo general, la gente no ve la información de paquetes comunes que pasan sólo para ver si hay algo interesante en ellos. Podrían esperar mucho tiempo para que alguien envíe de manera accidental un fragmento que contenga información interesante.

Si alguien de dentro usa intencionalmente la fragmentación para transmitir información, es que usted tiene usuarios hostiles dentro de su Firewall puede enfrentar exitosamente este problema (sin embargo, quizá no sean usuarios hostiles muy astutos por que hay maneras más sencillas de obtener la información.)

La única situación en la que debe preocuparse de que los fragmentos salgan es cuando permite entrar una solicitud pero bloque a la respuesta. En este caso, saldrán los fragmentos de la respuesta (excepto los iniciales), y el atacante tendrá razón en esperarlos y buscarlos. Puede superar esto si es cuidadoso al filtrar las preguntas de salida y no confiando en filtrar la salida de respuestas.

3.2.6 PROTOCOLOS ARRIBA DE IP

IP funciona como la base para una múltiples protocolos; los más comunes son TCP, UDP e ICPM. De hecho, estos son los únicos protocolos basados en IP que usted podrá usar fuera de los ámbitos de investigación.

También analizamos los Procedimientos de Llamadas Remotas (RCP) en esta sección, aunque, en el sentido estricto de la palabra, se basan en TCP y UDP y no en IP. No obstante, tiene sentido explicarlos aquí porque, al igual que TCP y UDP, RCP tiene la función de operar como un protocolo de sesión de propósito general sobre el que pueden descansar los protocolos de aplicación.

Además, analizamos brevemente IP sobre IP (por ejemplo, un paquete IP encapsulado dentro de otro paquete IP), lo cual se usa básicamente para paquetes IP tipo túnel multicast sobre redes IP que no son multicast.

TCP

TCP es el protocolo usado más comúnmente para los servicios de Internet. Por ejemplo, Telnet, FTP,SMTP, NNTP y HTTP son, todos, servicios basados en TCP. TCP proporciona una conexión bidireccional confiable en dos extremos. Abrir una conexión TCP es como hacer una llamada telefónica: usted marca el número y tras un breve periodo de espera, se establece una conexión lo suficientemente confiable con quien quiere hablar.

TCP es *confiable* porque da tres garantías al nivel de aplicación:

- El destino recibe la información de nivel de aplicación en el orden en que fue enviada
- El destino recibe toda la información de nivel de aplicación
- El destino no recibe duplicados de ninguna información de nivel de aplicación

TCP corta una conexión antes de violar una de tales garantías. Por ejemplo, si los paquetes TCP de la mitad de una sesión se pierden en su tránsito al destino, TCP hace que se retransmitan antes de pasar la información al nivel de aplicación. No pasará la información que sigue a la información faltante hasta tener esta última. Si alguna información no puede recobrarse, a pesar de intentos repetidos, el nivel TCP corta la conexión y lo reporta al nivel de aplicación, en lugar de pasarle la información con un faltante.

Estas garantías ocasionan cierto costo tanto en tiempo de espera (los dos lados de una conexión deben intercambiar datos de inicio antes de empezar a transferir la información) como de desempeño (ambos extremos deben llevar un control del estado de la conexión para determinar qué información debe enviarse otra vez al otro extremo para cubrir los faltantes en la conversación).

TCP se vuelve bidireccional una vez que se establece una conexión: un servidor puede contestar a un cliente sobre la misma conexión. No necesita establecer una conexión del cliente al servidor para solicitudes o comandos y otra del servidor al cliente para las respuestas a ellos.

Si intenta bloquear una conexión TCP, basta con detener el primer paquete. Sin él (y, aún más, sin la información de inicio que contiene para la conexión), el receptor no ensamblará ningún otro paquete de esa conexión en un flujo de información y la conexión no se establecerá. Se reconoce el primer paquete porque el bit ACK del encabezado TCP no está encendido; todos los demás paquetes de la conexión, sin importar en qué dirección viajen, tendrán encendido el bit ACK.

Reconocer estos paquetes de TCP de "inicio de conexión" le permite aplicar una política que permita que los clientes internos se conecten a servidores externos, pero que evite que clientes externos se conecten a servidores internos. Logrará esto permitiendo que los paquetes TCP de inicio de conexión (los que no tienen encendido el bit ACK) sólo salgan y no entren. Se permitirá que los paquetes de inicio de conexión salgan de los clientes internos a servidores externos, pero no se permitirá su entrada de clientes externos a servidores. Los atacantes no pueden alterar esta acción simplemente encendiendo el bit ACK en sus paquetes de inicio de conexión, ya que la ausencia de este bit es lo que identifica tales paquetes como de inicio de conexión.

Las implementaciones de filtrado de paquetes varían en cómo tratan y le permiten a usted manejar el bit ACK. Algunas dan acceso directo a él (por ejemplo, dejándole incluir "ack" como una palabra clave en una regla para el filtrado de paquetes). Algunas otras dan acceso indirecto a ese bit. Por ejemplo, la palabra clave "establecido" (established) de Cisco funciona examinándolo (establecido será "cierto" si el bit ACK está encendido y "falso" si no lo está). Por último, algunas implementaciones no le permiten examinar el bit ACK.

UDP

El cuerpo de un paquete IP puede contener un paquete UDP en lugar de un paquete TCP. UDP es una alternativa de bajo *overhead* para TCP.

UDP es de bajo *overhead* porque no brinda ninguna de las garantías de confiabilidad (envío, orden y no duplicado) que da TCP y, por lo tanto, no necesita el mecanismo para procesarlas. Cada paquete (datagrama) UDP es independiente. Los paquetes UDP no son parte de un "circuito virtual", como los de TCP. Enviar paquetes UDP es como dejar tarjetas postales en el correo: si deja cien postales ahí aunque todas tengan la misma dirección, no pueden tener la certeza de que todas llegarán a su destino, y las que lleguen quizá no estarán exactamente en el mismo orden en que fueron enviadas.

A diferencia de las tarjetas postales, los paquetes UDP se pueden hacer llegar más de una vez (sin ser destrozados, que es, por lo común, la única forma en que la misma postal se envíe varias veces). Son posibles múltiples copias porque el paquete puede ser duplicado por los niveles inferiores de red. Por ejemplo, en Ethernet, el paquete se duplicaría si un router pensara que pudo haber sido víctima de una colisión en Ethernet. Si el router está equivocado y el paquete original no sufrió una colisión, tanto el original como el duplicado llegarán finalmente a su destino (una aplicación confusa también podría decidir enviar la misma información dos veces, tal vez porque no obtuvo la respuesta esperada a la primera.).

A los paquetes TCP también puede pasarles esto, pero se corrigen antes de que se envíe la información al nivel de aplicación. Con UDP, este nivel es responsable de tratar con los paquetes, no con la información corregida.

En su estructura, los paquetes UDP son muy similares a los de TCP. Un encabezado UDP contiene números de puerto UDP fuente y destino, igual que los números de puerto TCP fuente y destino. Sin embargo, un encabezado UDP no contiene nada que se parezca a un bit ACK. El bit ACK es parte del mecanismo para garantizar la confiabilidad en el envío de la información. Como UDP no da tales garantías (es un protocolo no confiable), no necesita un bit ACK. No hay forma de que un router con filtrado de paquetes determine, con solo examinar el encabezado de un paquete UDP entrante, si es el primer paquete de un cliente externo a un servidor interno, o una respuesta de un servidor externo a un cliente interno.

Algunos usos del filtrado de paquetes, como el producto Firewall-1 de Checkpoint, Janus; SecureConnect Router de Morning Star; y KarlBridge/KarlBrouter, tienen la capacidad de "recordar" los paquetes UDP salientes que han visto. Así pueden permitir que regresen únicamente los paquetes de respuesta a través del mecanismo de filtrado. Para que se tome como respuesta, un paquete entrante debe ser del anfitrión y puerto al que se envío el paquete saliente y debe estar dirigido al anfitrión y puerto que envió el paquete de salida. Esta capacidad con frecuencia se denomina *filtrado dinámico de paquetes* , ya que el router modifica, esencialmente, las reglas para el filtrado al instante para acomodar estos paquetes que regresan. Las reglas creadas para permitir las respuestas son de tiempo limitado; su tiempo se vence después de ciertos segundos o minutos. El filtrado dinámico de paquetes también puede usarse para cualquier situación en la cual cambien las reglas para el filtrado de paquetes sin que alguien modifique la configuración en forma explícita; productos distintos soportan capacidades diferentes.

ICMP

ICMP se usa para mensajes de estado y control de IP. Los paquetes ICPM se envían en el cuerpo de los paquetes IP, igual que los paquetes TCP y UDP: Ejemplos de mensajes de ICMP incluyen:

- Solicitud de eco (Echo request), lo que envía un anfitrión cuando usted ejecuta ping
- Respuesta de eco (Echo response), con lo que un anfitrión responde a una solicitud de eco (ping).
- Tiempo excedido (Time exceeded), lo que devuelve un router cuando determina que un paquete parece estar ciclado; un nombre más intuitivo podría ser número máximo de saltos excedido
- Destino inalcanzable (Destination Unreachable), lo que devuelve un router cuando no se puede llegar al destino de un paquete por alguna razón (por ejemplo, porque un enlace de red está caído)
- Redirigir (Redirect), lo que un router envía a un anfitrión en respuesta a un paquete que aquel debió enviar a un router distinto; el router maneja el paquete original de cualquier forma (dirigiéndolo al router al que debió irse desde el inicio)y el mensaje de redirigir informa al anfitrión la ruta más eficiente para la próxima vez.

A diferencia de TCP o UDP, ICMP no tiene puertos o destino, y ningún otro protocolo sobre él. En vez de eso, hay grupo de códigos de ICMP definidos; el código específico utilizado dicta la interpretación del resto del paquete ICMP.

Muchos sistemas para filtrado de paquetes le permiten filtrar paquetes ICMP basados en el campo Tipo de mensaje, casi igual que como le permiten filtrar paquetes TCP o UDP basados en los campos de puerto fuente y destino TCP o UDP.

RPC

Hay múltiples protocolos del tipo procedimientos de llamadas remotas conocidos como RCP. El más popular se conoce a veces como "Sun RPC", debido a que fue originalmente creado por Sun Microsystems. Este es el protocolo que analizaremos y es al que con más frecuencia sólo se le llama "RPC". Otros mecanismos de procedimientos de llamadas remotas son específicos para implementaciones de UNIX o familias de implementaciones específicas en UNIX (por ejemplo, OSF DCE tiene su propio protocolo de procedimiento de llamada remota). Estos mecanismos difieren en el detalle del RCP, pero tienden a tener problemas similares.

En este sentido estricto, el mecanismo RPC no está construido arriba de IP sino de UDP y TCP. Sin embargo, al igual que TCP y UDP, una gran variedad de protocolos de aplicación (como NFS y NIS/YP)usan RPC como protocolo de propósito general. NFS y NIS/YP son servicios vulnerables desde el punto de vista de la seguridad de una red. Un atacante con acceso al servidor NIS/YP puede obtener el archivo de contraseñas, sobre el que puede ejecutar un ataque de ruptura de contraseñas contra su sistema.

En los protocolos TCP y UDP, los números de puerto son campos de dos bytes, lo cual significa que ha sólo 65 536 números de puertos posibles para los servicios TCP y UDP. No hay puertos suficientes para poder asignar un número de puerto único para cada servicio y aplicación posible que quisiéramos tener. Entre otras cosas, RPC supera esta limitante. A cada servicio basado en RPC se le asigna un "número de servicio RPC" único de cuatro bytes, lo cual posibilita que 4 294 967 296 servicios tengan un número único, calidad más que suficiente para asignar un número único a cada servicio y aplicación posible.

RPC está construido sobre TCP y UDP, así que ahí debe haber un modo de *mapear* los números de servicio RPC de los servidores basados en RPC que estén usándose en una máquina, a los puertos TCP o UDP específicos que sus servidores están usando. Es aquí donde entra el servidor *portmapper* (localizador de puerto).

Portmapper es el único servidor relacionado con RPC que está garantizado para ejecutarse en un número de puerto TCP o UDP específico (el número de puerto 111 en ambos casos). Cuando se inicia un servidor basado en RPC, como el servidor NFS o NIS/YP, asignan un puerto TCP y/o UDP aleatorio (algunos usan uno, algunos el otro, algunos ambos) para sí mismo. De este modo, contacta al servidor *portmapper* en la misma máquina para "registrar" su número único de servicio y el (los) puerto(s) que usa en ese momento.

Un programa cliente basado en RPC que quiere establecer contacto con servidor específico basado en RPC en una máquina, primero contacta al servidor *portmapper* en esa máquina (el cual, recuerde, siempre se ejecuta en el puerto 111, tanto de TCP como de UDP). El cliente le dice al *prtmapper* el número de servicio RPC único del servidor al que quiere acceder; el *portmapper* responde con un mensaje que dice: "Lo siento, pero ese servicio no está disponible en esta máquina por el momento", o "Ese servicio se ejecuta en este momento en el puerto TCP (o UDP)*n* de esta máquina".

En ese punto, el cliente se pone en contacto con el servidor en el número de puerto que obtuvo del *portmapper* y continua su conversación directamente con él, sin mayor relación con el *portmapper*.

Es muy difícil usar el filtrado de paquetes para controlar servicios basados en RPC, pues usted no sabe qué puerto usará el servicio en una máquina específica, y existe la probabilidad de que el puerto usado cambie cada vez que la máquina vuelva a arrancar. No es suficiente bloquear el acceso al *portmapper*. Un atacante puede saltar el paso de comunicarse con él y simplemente probar en todos los puertos TCP y/o UDP (los 65 536 puertos posibles pueden revisarse en una máquina común en cuestión de minutos), para buscar la respuesta esperada de un servidor específico basado en RPC, como NFS o NIS/YP.

Algunos productos más novedosos para filtrado de paquetes pueden comunicarse con el *portmapper* para determinar dónde están ciertos servicios y filtrar con base en eso. Observe que esto debe ser verificado por paquete para servicios basados en UDP. El filtro de paquetes tendrá que ponerse en contacto con el *portmapper* cada vez que reciba un paquete, porque si la máquina ha vuelto ha arrancar, el servicio puede haber cambiado de puerto. Como TCP está orientado a conexión, el número de puerto sólo debe verificarse una vez por conexión. Emplear este mecanismo para permitir servicios basados en UDP redundará en una gran cantidad de tráfico y quizá no sea prudente usarlo para aplicaciones de información intensa, como NFS.

NOTA

Aunque no es suficiente, de todos modos debe bloquear el acceso al portmapper, ya que algunas versiones de éste pueden emplearse como proxy para los clientes de un atacante. ¿qué puede hacer entonces para proteger los servicios basados en RPC? Un par de observaciones: primero, resulta que la mayoría de los servicios "peligrosos" basados en RPC (en especial NIS/YP y NFS)sólo ofrecen sobre UDP. Segundo, la mayoría de los servicios a los que querría entrar a través de un filtro de paquetes se basan en TCP, no en UDP; las excepciones notables son DSN, NTP, syslog y Archie. Estas dos observaciones llevan al planteamiento común de muchos sitios cuando se ven frente a RPC y que usan filtrado de paquetes: bloquear todo UDP, excepto los puertos específicos y fuertemente controlados de DNS, NTP, syslog y Archie.

Con este enfoque, si desea permitir algún servicio RPC basado en TCP, deberá permitirlos todos. Los servidores NFS basados en TCP, aunque disponibles, aún no son de uso generalizado; sin embargo, si usted los utiliza, debe modificar este enfoque según se requiera.

3.2.7 IP SOBRE IP

En algunas circunstancias, los paquetes IP están encapsulados dentro de otros paquetes IP para su transmisión, de ahí que se llamen "IP sobre IP". El uso más común de IP sobre IP es para llevar paquetes IP *multicast* (es decir, paquetes con direcciones destino multicast)entre redes que soportan multicast sobre redes intermedias que no lo hacen. Para cruzar estas redes intermedias, un router especial de multicast (*o mrouter*)en cada red multicast encapsula los paquetes IP multicast que quiere enviar dentro de paquetes IP unicast (es decir, normales) dirigidos a otros *mrouters* . Los otros *mrouters*, al recibir estos paquetes multicast encapsulados, quitan el paquete exterior (unicast) y después manejan el interior (multicast).

El multicast IP está volviéndose más y más popular en Internet, sobre todo por el servicio de conferencias y otros servicios ofrecidos a través de MBONE.

3.2.8 PROTOCOLOS DEBAJO DE IP

Teóricamente es posible filtrar información debajo del nivel IP (por ejemplo, la dirección de hardware de Ethernet). Sin embargo, hacerlo es poco útil, ya que en la mayoría de los casos, todos los paquetes del exterior vienen de la misma dirección de hardware (la dirección del router que maneja su conexión con Internet). Más aún, muchos routers tiene múltiples conexiones con diferentes protocolos de nivel inferior. Usted no podía escribir una regla que se aplicara a todas las interfaces de un router que tiene dos conexiones Ethernet y una conexión FDDI, ya que aunque los encabezados Ethernet y FDDI son similares, no son idénticos. En la práctica, IP es el protocolo de nivel más bajo que la gente elige para filtrar paquetes.

3.2.9. PROTOCOLOS DE NIVEL DE APLICACIÓN

En la mayoría de los casos, hay un protocolo más arriba de TCP o UDP específico a la a aplicación. Estos protocolos difieren ampliamente en su especificidad, y hay cientos de ellos, si no es que miles (casi tantos como aplicaciones basadas en red). Algunas aplicaciones más modernas para filtrado de paquetes proporcionan la habilidad de filtrar en protocolos de nivel de aplicación para aplicaciones muy conocidas. Por ejemplo, pueden ser capaces de reconocer información específica en una operación FTP para instalar filtros dinámicos, o pueden comparar la información de un paquete con la aplicación a la que se supone que está dirigido, a fin de tener la certeza de que los paquetes dirigidos a un puerto DNS efectivamente son paquetes DNS.

3.2.10 IP Versión 6

La versión actual de IP se conoce, de modo oficial, como IP versión 4; cuando a lo largo de este trabajo hablamos de IP sin mayor calificación, nos referimos a esa versión. Sin embargo, hoy en día existe una versión nueva en desarrollo, conocida como IP versión 6 (Ipv6 para abreviar). ¿Por qué necesitamos una versión de IP y cómo afectará IPv6?

Lo que motivo la creación de Ipv6 fue un sencillo problema: Internet está quedándose sin direcciones IP. Esa red se ha vuelto tan popular que ya no habrá espacio para que sigan asignando números de red IP (sobre todo números de red clase B, los cuales, se ha comprobado, son los que necesita la mayoría de los sitios); se calcula que, si nada se hubiera hecho, Internet se habría quedado sin direcciones en 1995 o 1996. Por fortuna, el problema se detectó y se hizo algo. De hecho, se hicieron dos cosas; primero, la implementación de un conjunto de medidas y guías temporales para hacer el mejor uso posible de la direcciones que quedaban sin asignar; segundo, se diseñó e implementó una versión de IP que afrontara, de modo permanente, el asunto de las direcciones agotadas.

Si usted va a crear una versión de IP para hacer frente a la falta de espacio de direcciones, también debe aprovechar la oportunidad de tratar con todo un conjunto de otros problema o limitantes, tales como la encriptación, la autenticación, el enrutamiento fuente y la configuración dinámica. Según Steve Bellovin, de los laboratorios Bell de AT&T, un experto en Firewalls de renombre en Internet y participante en el proceso de diseño de IPv6:

IPv6 se basa en el concepto de encabezados variables. Así es como se hacen la encriptación y la autenticación; el campo "next protocol"(siguiente protocolo)después del encabezado de IPv6 especifica un encabezado de encriptación o de autenticación. Por lo tanto, sus siguientes campos de protocolo por lo general indicarían ya sea protocolos IPv6 o uno de los protocolos de transporte usuales, como TCP o UDP. Puede incorporarse IP sobre IP aun sin encriptación o autenticación; eso puede usarse como una forma de enrutamiento fuente. Un método más eficiente es usar el encabezado con enrutamiento fuente, que es más útil que la opción correspondiente de IPv4 y acaso se use más, en especial para IP móvil.

Algunas de las implicaciones para los Firewalls ya son aparentes. Un filtro de paquetes debe estar al final de una larga cadena de encabezados, entendiendo y procesando cada uno a la vez. (Por lo tanto, esto puede hacer más caro ver los números de puerto.)Una postura más apropiada y cautelosa dicta que un paquete con un encabezado desconocido debe ser rebotado, ya sea de entrada o de salida. Además, la facilidad y frecuencia del enrutamiento fuente significa que la autenticación criptográfica es absolutamente necesaria. Por otro lado, si intenta que tal autenticación sea un elemento estándar y obligatorio. Los paquetes encriptados son opacos y, por lo tanto, no se pueden examinar; esto es cierto hoy en día, por supuesto, pero no hay muchos encriptadores en uso en la actualidad, lo cual cambiará. También observe que puede hacerse encriptación de anfitrión a anfitrión, de anfitrión a compuerta, o de compuerta a compuerta, lo que complica aún más el análisis.

El filtrado basado en la dirección también se vera afectado, en algún grado, por los nuevos mecanismos de autoconfiguración. Es importante que cualquier anfitrión cuya dirección se mencione en un filtro reciba la misma dirección cada vez. Aunque este es el propósito de los mecanismos estándares, hay que ser cuidadoso con los esquemas propietarios, servidores para acceso conmutado, etcétera. También pueden cambiar los bits de dirección mayor orden, para lograr un esquema de direccionamiento basado en el proveedor y un intercambio sencillo entre grandes carriers.

Finalmente, IPv6 incorpora los "flujos", circuitos esencialmente virtuales en el nivel IP, se han destinado para emplearlos en elementos como video, selección de saltos intermedios de circuitos ATM, etcétera. Pero si se les da una autenticación apropiada, también pueden usarse para los Firewalls: se acabaría el problema de la respuesta UDP si la pregunta tuviera una identificación tipo flujo referida por la respuesta. Por cierto, esta es una vaga idea mía; no hay estándares de cómo debe hacerse esto. El protocolo común establecido para el flujo no funcionará; es demasiado caro. Pero un encabezado transversal de Firewall podría funcionar.

Como puede ver, IPv6 podría tener gran impacto en los Firewalls, en especial con respecto al filtrado de paquetes. Tenga en cuenta que Ipv6 no va a aparecer de pronto. Ipv4 permanecerá un largo tiempo y muchos sitios continuarán ejecutándolo en el futuro cercano. Los diseñadores de IPv6 son muy sensibles a los asuntos de transición y ponen mucha atención en esa área y en varias estrategias de migración que podrían utilizar los sitios.

3.2.11 PROTOCOLOS NO IP

Otros protocolos en el mismo nivel que IP, como AppleTalk e IPX, dan tipos de información similar a la de IP, aunque sus encabezados y operaciones y, por lo tanto, sus características de filtrado de paquetes, varían radicalmente. Muchos usos de filtrado de paquetes soportan sólo filtros IP y bloquean tráfico no IP.

Algunos paquetes dan soporte limitado para filtrado de paquetes, pero por lo general el soporte es mucho menos flexible y de menor capacidad que las capacidades de filtrado de un router IP.

De momento, el filtrado de paquetes como herramienta no es tan popular y bien desarrollada para protocolos no IP, quizá porque éstos casi no se utilizan para comunicarse fuera de una solo organización sobre Internet (Internet es, por definición, una red de redes IP). Los protocolos no IP son más un elemento para los Firewalls internos de una organización y, para esta aplicación, debe elegir uno de los programas que soportan filtros no IP.

A lo largo de Internet, los protocolos no IP se manejan encapsulados dentro de los protocolos IP. En muchos casos, se le limitará a permitir o prohibir protocolos encapsulados en su totalidad: puede aceptar las conexiones Appletalk en UDP, o rechazarlas del todo. Algunos programas que soportan protocolos no IP pueden reconocer estas conexiones cuando están encapsuladas en sus campos.

3.2.12 QUÉ HACE EL ROUTER CON LOS PAQUETES

Una vez que un router con filtrado de paquetes ha terminado de examinar un paquete específico, ¿qué puede hacer con él. Hay dos alternativas:

- Pasar el paquete. Por lo común, si éste pasa el criterio de la configuración para filtrado de paquetes, el router lo envía a su destino, como haría un router normal (no un router con filtrado de paquetes)
- Rechazar el paquete. La otra alternativa obvia es rechazar el paquete si no pasa el criterio de la configuración para filtrado de paquetes

3.2.13 REGISTRO DE LAS ACCIONES

Sin importar si el paquete pasa o es eliminado ("permitido" o "rechazado"), tal vez desee que el router registre la acción tomada. Esto es cierto en especial si elimina el paquete, porque queda fuera de sus reglas para filtrado de paquetes. En ese caso, usted querrá saber qué se intentó y no se permitió.

Quizá no va a registrar cada paquete permitido, pero sí algunos. Por ejemplo, tal vez quiera registrar los paquetes TCP de inicio de conexión para dar seguimiento a las conexiones TCP que entran y salen del sistema. No todos los filtros de paquetes registran los paquetes permitidos.

Distintas implementaciones del filtrado de paquetes soportan diferentes formas de registro. Algunas registran sólo información específica de un paquete; otras envían o registran todo el paquete. Por lo general, su filtro de paquetes debe configurarse para hacer los registros en algún anfitrión por medio del servicio *syslog*. No querrá que la única copia de las claves de acceso estén en el filtrado de paquetes si éste se ve amenazado. La mayor parte del filtrado de paquetes también ocurren en routers dedicados, los cuales rara vez tienen grandes cantidades de espacio en disco para dedicarlo al registro de información.

3.2.14 FILTRADOS POR INTERFACE

Hay una pieza clave de información útil cuando toma una decisión de filtrado de paquetes, que no pueden encontrase en los encabezados de paquete; es la interface por la cual el paquete entró al router o saldrá de él. Es una información importante que permite que el router detecte paquetes falsos. Si el único router entre su red interna y el mundo exterior recibe un paquete con una dirección fuente interna de la interface interna, no hay problema; todos los paquetes que vienen de adentro tendrán direcciones fuente internas. Sin embargo, si el router recibe un paquete con una dirección fuente interna de la interface externa, significa que alguien está falsificando el paquete (probablemente en un intento por burlar la seguridad, o que hay algo que anda muy mal en la configuración de su red. Puede obtener estos paquetes sin falsificación. Por ejemplo, alguien podría tener una segunda conexión entre su red y el mundo exterior, como una conexión telefónica PPP, probablemente con muy poca o ninguna relación con la seguridad. Como resultado, el tráfico que debía permanecer interno a su red está "goteando"al exterior a través de esta segunda conexión, pasando a través de Internet (e intentando regresar a través de su "puerta principal"). Hay poco que pueda hacer para detectar tales conexiones ilícitas " de puerta trasera", excepto rastrear los paquetes internos que llegan de fuera; lo mejor que puede hacer es tener una política estricta y bien divulgada contra ellos así como proporcionar la mayor cantidad posible de los servicios que sus usuarios deseen a través de la puerta principal (el Firewall), para que no sientan la necesidad de crear su propia puerta trasera. A estos paquetes debe darles acceso y tratarlos como asuntos urgentes. Si alguien los está falsificando, esa persona lo está atacando seriamente. Si los paquetes entran por una puerta trasera, tiene un problema de seguridad a causa de la conexión adicional con Internet. También puede tener un problema de enrutado: un anfitrión que dice ser interno y advierte que hay rutas para sí mismo está en peligro de meterse en todo tráfico de su red interna. Esto es malo si es una conexión PPP, que tal vez no podrá manejar la carga. Es peor si no está conectado del todo a su red, ya que parte o todo el tráfico de su red va a desaparecer.

3.2.15 DEVOLUCIÓN DE CÓDIGOS DE ERROR DE ICMP

Si se va a eliminar un paquete, el router puede o no enviar un código de error de ICMP indicando lo que pasó. Enviar como respuesta un código de error de ICMP tiene el efecto de prevenir la máquina emisora para que no vuelva a enviar el paquete; por lo tanto, ahorra algo de tráfico en la red y algún tiempo al usuario en el extremo remoto (si envía de nuevo un código de error de ICMP, el intento de la conexión del usuario fallará de inmediato; de otro modo, su tiempo expiará, lo que puede tomar varios minutos).

Hay dos conjuntos importantes de códigos de ICMP para escoger:

- Los códigos genéricos "destino inalcanzable" y "red inalcanzable"

- Los códigos "destino administrativamente inalcanzable" y, en particular, los códigos "anfitrión administrativamente inalcanzable" y "red administrativamente inalcanzable"

Los diseñadores de ICMP pensaron que el primer par de códigos de error de ICMP que devolviera el router, "anfitrión inalcanzable" o "red inalcanzable", indicaran problemas serios de la red: el anfitrión destino no está en operación o hay algo sin operar en la única ruta de acceso a él. Estos códigos de error preceden a los Firewalls y al filtrado de paquetes. El problema de devolver uno de estos códigos de error es que algunos servidores (sobre todo cuando se están ejecutando versiones antiguas de UNIX) los toman demasiado literalmente. Si estas máquinas reciben un código de "anfitrión inalcanzable" para un anfitrión dado, supondrán que el anfitrión es del todo inalcanzable y cerrarán todas las conexiones fueron permitidas por el filtrado de paquetes.

El segundo grupo de códigos de error de ICMP que puede devolver el router, "anfitrión administrativamente inalcanzable", se agregaron hace unos años a la lista oficial de tipos de mensajes de ICMP, en especial para dar a los sistemas de filtrado de paquetes algo para indicar cuando desechaban un paquete. Muchos sistemas aún no reconocen estos códigos de error de ICMP que no entienden, así que esto debe ser equivalente a no enviar un código de error a tales sistemas. Por otro lado, muchos dispositivos cumplen de manera deficiente con los estándares y esta es la clase de condición limitante que algunos de ellos no podrían manejar en forma adecuada. El hecho de que el estándar indique que un sistema ignore los códigos de error no conocidos, no asegura que un sistema no reaccionará en forma desastrosa ante tal error. ¡Simplemente asegura que podrá alegar persuasivamente que no es su culpa si lo hace!

Hay varios elementos que debe considerar cuando decida si su sistema para filtrado de paquetes va a devolver o no códigos de error de ICMP.

- ¿Qué mensaje debe enviar?
- ¿Puede afrontar los gastos de generar y devolver códigos de error?
- ¿Devolver esos códigos permitirá a los atacantes obtener demasiada información sobre el filtrado de paquetes?

¿Qué grupo de mensajes de error funciona para su sitio?

Devolver los viejos códigos "anfitrión inalcanzable" y "red inalcanzable" es técnicamente incorrecto (recuerde que el anfitrión puede o no ser inalcanzable, de acuerdo con la política de filtrado de paquetes, dependiendo de qué anfitrión intenta acceder qué servicio). Además, estos códigos de error pueden ocasionar que muchos sistemas reaccionen exageradamente (cerrando todas las conexiones con ese anfitrión o red).

Devolver los códigos nuevos, "anfitrión administrativamente inalcanzable" o "red administrativamente inalcanzable", anuncia el hecho de que hay un sistema para filtrado de paquetes en su sitio, algo que usted puede querer hacer o no. Estos códigos también ocasionan reacciones excesivas cuando se utilizan implementación IP con errores.

También hay otra consideración. Generar y devolver códigos de error de ICMP requiere cierta cantidad de esfuerzo de parte del router con filtrado de paquetes. Un atacante podría, quizá, montar un ataque de negación del servicio inundado al router con paquetes que éste rechazaría y para los cuales intentaría generar paquetes de error de ICMP.

El problema no es el ancho de la banda en la red; es la carga de CPU en el router (mientras está ocupado generando paquetes ICMP, no puede hacer otras tareas tan rápido, como hacer decisiones de filtrado). Por otro lado, no devolver códigos de error de ICMP ocasionará una pequeña cantidad de tráfico excesivo en la red, mientras el sistema para filtrado de paquetes debe ser una fracción del número total de paquetes procesados (si no es una pequeña fracción, tiene problemas más serios, porque, al parecer, la gente intenta muchas cosas que "no están permitidas").

Si su router devuelve un código de error de ICMP para cada paquete que viola su política de filtrado, también le está dando al atacante un modo de probar sus sistema de filtrado. Al observar qué paquetes generan una respuesta de error de ICMP, los atacantes pueden descubrir que qué tipos de paquetes violan y cuáles no violan su política (por consiguiente, qué tipos de paquetes se permiten y cuáles no se permiten dentro de su red). No debe dar esta información porque simplifica mucho la tarea del atacante, quien sabe que los paquetes que no generan mensaje de error de ICMP van a algún lado, y puede concentrase en esos protocolos, donde de hecho usted es vulnerable. Será preferible que el atacante pase mucho tiempo enviándole paquetes que usted felizmente tirará. Devolver códigos de error de ICMP acelera los programas de ataque; si obtienen un mensaje de error de ICMP para algo que intentan, no tienen que esperar a que el tiempo expire.

Después de todo, lo más seguro parece que es desechar los para los paquetes sin devolver códigos de error de ICMP. Si su router ofrece la flexibilidad suficiente, tiene sentido configurarlo para que devuelva códigos de error a sistemas internos (que querrían saber de inmediato que algo va a fallar en vez de esperar a que el tiempo expire), pero no a sistemas externos (donde la información le daría un atacante un modo de probar la configuración del filtrado del Firewall). Aun si su router no parece ofrecer tal flexibilidad, puede obtener el mismo resultado especificando a las reglas del filtrado de paquetes que permitan los paquetes ICMP pertinentes de entrada, y no permitan los de salida.

3.2.16 CONVENCIONES PARA LAS REGLAS DE FILTRADO DE PAQUETES

Hay algunos temas que debe saber sobre estas reglas.

Para evitar confusiones, en la medida de lo posible las reglas utilizadas como ejemplo se especifican con descripciones abstractas, en lugar de hacerlo con direcciones reales. En vez de usar direcciones reales fuente y destino (digamos 172.16.51.50), usamos las palabras "Interna" o "Externa" para identificar de qué redes estamos hablando. Por lo común, los sistemas reales de filtrado de paquetes requieren que especifique en forma explícita los rangos de dirección; la sintaxis varía de un router a otro.

En todos nuestros ejemplos de filtrado de paquetes se supone que, por cada paquete, el router revisa las reglas en orden hasta que encuentra la que coincide y después ejecuta la acción específica por ella. Suponemos una "negativa" implícita predeterminada en el sistema si no se aplican las reglas, aunque es buena idea especificarlo de manera explícita (y generalmente lo hacemos).

La sintaxis usada en nuestros ejemplos de filtrado específica el número de bits significativos que se utilizan para compararlos con otras direcciones después de un carácter de separación (/). Por eso, 10.0.0.0/8 se acopla con cualquier dirección que empiece con 10; es equivalente a 10.0.0.0 con una máscara de una red de 255.0.0.0 o 10.0.0.0 con una máscara comodín de 0.255.255.255, o (si fuera un nombre de archivo) 10.*.*.*.

Aunque intentamos ser lo más específicos posible en estos ejemplos, es imposible decirle con precisión qué debe especificar en su producto para filtrado de paquetes. El mecanismo exacto para establecer las reglas para filtrado de paquetes varía mucho de un producto a otro. Algunos productos (como screend) le permiten especificar un solo grupo de reglas que se aplican a todos los paquetes enrutados por el sistema. Otros (como Telebit NetBlazer) le permiten establecer reglas para interfaces específicas. Hay otros (como los productos Livingston y Cisco) que le permiten especificar grupos de reglas y después aplicarlos por nombre a interfaces específicas (para que, por ejemplo, pueda definir un conjunto de reglas compartido por un número de interfaces diferentes, e instalar en un grupo distinto las reglas que son únicas a una interface dada).

He aquí un sencillo ejemplo para ilustrar las diferencias. Elegimos tres sistemas porque de alguna manera representan los distintos modos de especificar filtros, no por ninguna preferencia particular hacia ellos; en general, otros sistemas son similares a estos. Por ejemplo, los productos de Cisco se parecen a los de Livington en que crean conjuntos de reglas, que luego se aplican a los paquetes que van en una dirección específica a través de una interface específica. Son distintos en detalles de sintaxis, tales como la forma en que usted especifica las direcciones de los servidores y las máscaras de bit.

3.2.17 CONSEJOS Y TRUCOS PARA FILTRADO DE PAQUETES

Por lo general son mínimas las herramientas de edición de filtros en la mayoría de los sistemas. Además, no siempre resulta evidente cómo interactúan las nuevas reglas con los conjuntos de éstas ya existentes. En particular, con frecuencia es difícil eliminar las reglas o agregar reglas a la mitad de un conjunto existente. Puede serle más conveniente guardar sus filtros en un archivo de texto en uno de sus sistemas UNIX o de sus computadoras personales para editarlas ahí con las herramientas que conoce mejor y después cargar el archivo en el sistema para filtrado como si tuviera comandos que estuviera tecleando en la consola. Distintos sistemas soportan varias formas de hacerlo. Por ejemplo, en los productos Cisco puede usar TFTP para obtener archivos de comandos de un servidor TFTP. Los productos Livingston contienen un programa disponible llamado pmcommand que descarga comando al equipo. Otros productos tienen otros mecanismos. Una ventaja adicional de resguardar los filtros en otro lado, digamos un archivo, es que puede guardar comentarios en éste (cortándolos de la copia que envía al router, si es necesario). La mayoría de los sistemas para filtrado que eliminan cualquier comentario en los comandos que reciben; si regresa después a ver los filtros activos en el sistema, verá que no se conservaron los comentarios. RECARGAR LOS GRUPOS DE REGLAS DESDE EL INICIO CADA VEZ Lo primero que debe hacer el archivo es quitar las reglas viejas para que cada vez que usted lo cargue, reconstruya el conjunto de reglas desde el inicio; así, no tiene que preocuparse por cómo va a interactuar las nuevas reglas con las viejas. Después, especifique las reglas que quiere establecer, seguidas de los comandos necesarios para aplicarlas a las interfaces apropiadas. SIEMPRE USE DIRECCIONES IP; NUNCA NOMBRES DE ANFITRIÓN, siempre especifique los servidores y redes en las reglas para filtrado por dirección

IP, nunca por nombre de anfitrión o nombre de red (si es que su producto de filtrado lo soporta). Si especifica reglas para filtrado por nombre de anfitrión, su filtrado podría ser alterado si alguien corrompe, accidental o intencionalmente, la traducción del nombre a la dirección (por ejemplo, alimentando con información falsa a su servidor DNS).

Digamos que quiere permitir todo el tráfico entre un anfitrión externo de confianza (anfitrión 172.16.51.50) y servidores de su red interna (red clase C 192.168.10.0). En nuestros ejemplos, mostraríamos este caso de la siguiente forma:

Regla	Dirección	Dirección fuente	Dirección destino	ACK encendido	Acción
A	Entrada	Anfitrión externo confiable	Interna	Cualquiera	Permitir
B	Salida	Interno	Anfitrión externo confiable	Cualquiera	Permitir
C	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Prohibir

Con *screend,* especificaría:

between host 172.16.51.50 and net 192.168.10 accept ;
between host any and host any reject ;

Con Telebit NetBlazer, también tiene que especificar a qué regla de interface se aplicará, y si la regla se aplica a paquetes que entran o salen de esa interface. Para una interface externa llamada "syn0", sus reglas serían:

Permit 172.16.51.50/32 192.168.10/24 syn0 in
Deny 0.0.0.0/0 0.0.0.0/0 syn0 in

Permit 192.168.10/24 172.16.51.50/32 syn0 out
Deny 0.0.0.0/0 0.0.0.0/0 syn0 out

En Livingston PortMaster o IRX especificaría las reglas como un conjunto y después aplicaría el conjunto pertinente en la dirección correcta en la interface correcta. Si su interface externa se llama "s1", sus reglas se verían más o menos así:

Add filter s1.in
Set filter s1.in 1 permit 172.16.51.50/32 192.168.10.0/24
Set filter s1.in 2 deny 0.0.0.0/0 0.0.0.0/0
Set s1 ifilter s1.in

Add filter s1.out
Set filter s1.out 1 permit 192.168.10.0/24 172.16.51.50/32
Set filter s1.out 2 deny 0.0.0.0/0 0.0.0.0/0

Set s1 ofilter s1.out

En un router Cisco, usted también especifica las reglas como conjuntos y aplica los pertinentes en la dirección correcta en la interface externa se llama "serial1", sus reglas se verían así:

Access-list 101 permit ip 172.16.51.50 0.0.0.0 192.168.10.0 0.0.0.255
Access-list 101 deny ip 0.0.0.0 255 255.255.255 0.0.0.0 255.255.255.255
Interface serial 0
Access-group 101 in

Access-list 102 permit ip 192.168.10.0.0.0.0.255 172.16.51.50 0.0.0.0
Access-list 102 deny 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
Interface serial 0
Access-group 102 out

Para información detallada de la sintaxis de un paquete o producto en particular, consulte la documentación de ese paquete o producto. Una vez que la entienda, la sintaxis del sistema que usa, no debe tener mucha dificultad para trasladarse de nuestras tablas a la sintaxis de ese sistema.

NOTA

Observe las acciones predeterminadas implícitas. Los distintos sistemas para filtrado tienen diferentes acciones predeterminadas que se ejecutan si un paquete no se acopla con ninguna de las reglas de filtrado especificadas. Algunos sistemas prohiben todos los paquetes como esos. Otros sistemas ejecutan, como valor predeterminado, lo opuesto a la última regla; esto es, si la última regla fue un "permiso", el valor predeterminado del sistema es "prohibir", y si la ultima regla fue "prohibir", el valor predeterminado es "permitir". En cualquier caso, es buena idea establecer una regla explícita como valor predeterminado al final de sus reglas para filtrado de paquetes, a fin de que no tenga que preocuparse (o recordar siquiera) qué valor predeterminado usará su sistema.

3.2.18 FILTRADO POR DIRECCIÖN

La más simple, aunque no la más común, forma de filtrado de paquetes es el filtrado por dirección. Filtrar de ese modo le permite restringir el flujo de paquetes basándose en la dirección fuente y/o destino de los paquetes, sin tener que considerar qué protocolos están involucrados. Tal tipo de filtrado puede emplearse, por ejemplo, para permitir que ciertos servidores externos conversen con ciertos servidores internos, o para evitar que un atacante inyecte paquetes falsificados (paquetes manufacturados para que parezca que vienen de algún lugar distinto a su verdadera fuente) dentro de su red.

Digamos que quiere bloquear los paquetes que entran con direcciones fuente falsificadas; debe especificar esta regla:

Regla	Dirección	Dirección fuente	Dirección destino	Acción
A	Entrada	Interna	Cualquiera	Prohibir

Observe que dirección se refiere a su red interna. En el router entre su red interna e Internet, podría aplicar una regla de entrada ya sea a paquetes que entran en la interface del Internet o que salen en la interface interna; de cualquier modo, obtendrá el mismo resultado para el anfitrión protegido. La diferencia está en lo que ve el propio router. Si filtra paquetes de salida, el router no se está protegiendo a sí mismo.

3.2.19 RIESGOS DEL FILTRADO POR DIRECCIÓN FUENTE

No es necesariamente seguro confiar en las direcciones fuente, ya que pueden ser falsificadas. A menos que use alguna clase de autenticación criptográfica entre usted y el anfitrión con el quiere conversar, no sabrá si en verdad se comunica con él o con otra máquina que finge ser anfitrión. Los filtros que hemos explicado antes le ayudarán si un anfitrión externo finge ser un anfitrión interno, pero no harán nada con un anfitrión externo que diga ser un anfitrión externo distinto.

Hay dos tipos de ataques que se basan en la falsificación: *dirección fuente y hombre en el camino.*

En un ataque básico de falsificación de *dirección fuente*, un atacante envía sus paquetes que dicen ser de alguien en quien usted confía de algún modo, esperando que usted tome alguna acción basado en esa confianza, sin esperar obtener ningún paquete de regreso de parte de usted. Si el atacante no le importa obtener paquetes de regreso de su parte, no necesita estar en la ruta de acceso entre usted y quien quiera que éste fingiendo ser; puede estar en cualquier parte.

De hecho, sus respuestas irán a quien sea que el atacante finja ser, no al atacante. Sin embargo, si el atacante puede predecir sus respuestas, no necesita verlas. Muchos protocolos (si no la mayoría) son lo suficientemente predecibles para que un atacante habilidoso tenga éxito en esto. Hay muchos ataques que pueden ser llevados a cabo sin que el atacante necesite ver los resultados en forma directa. Suponga que un atacante manda un comando a su sistema que hace que usted le envíe a él su archivo de contraseñas por correo electrónico; si su sistema va a enviarle por correo su archivo de contraseñas, no necesita verlo durante el ataque en sí.

En muchas circunstancias (sobre todo las que tienen que ver con conexiones TCP), la máquina verdadera (la que finge ser el atacante) reaccionará a sus paquete (paquetes que intentan llevar a cabo una conversación de la que no sabe nada) intentado volver a establecer la conexión falsa. Obviamente, el atacante no quiere que esto pase. Debe asegurarse de que el atacante está completo antes de que la verdadera máquina obtenga los paquetes que está enviando, o antes de que obtenga los paquetes de fin de sesión de la máquina verdadera. Hay una variedad de modos para asegurarse de esto, por ejemplo:

- Realizar el ataque cuando la máquina verdadera está fuera de servicio
- Detener la máquina verdadera de tal forma que se pueda realizar el ataque
- Inutilizar la máquina verdadera mientras se realiza el ataque
- Confundir el router entre la máquina verdadera y el objetivo

- Usar el un ataque doble sólo se requiera la primera respuesta del paquete, de modo que no importe la terminación de la sesión

Los ataques de falsificación *hombre en el camino* depende de ser capaz de establecer una conversación completa afirmando ser un anfitrión de confianza. Para hacerlo, la máquina atacante debe ser capaz no sólo de enviarle paquetes sino, también, de interceptar los paquetes con los que usted responde. Para lograrlo, el atacante debe tomar una de las siguientes acciones:

- Insinuar a su máquina de ataque dentro de la ruta de acceso entre la máquina de usted y la máquina verdadera. Esto es más fácil de hacer cerca de los extremos de la ruta de acceso, y más difícil en alguna parte intermedia, ya que dada la naturaleza de las redes IP modernas, la ruta de acceso puede cambiar en cualquier segundo.
- Alterar la ruta de acceso entre las máquinas para que se dirija a la máquina atacante. Esto puede ser muy fácil o muy difícil, dependiendo de la topología de la red y del tipo de router usado por la red de usted, la red remota y los proveedores de servicio de Internet entre esas redes

Aunque esa clase de ataque se llama "hombre en el camino", es relativamente raro que en verdad se realice en el "camino", externo a los sitios en cada extremo porque nadie, excepto un proveedor de red, está en posición de realizarlo de ese modo, los proveedores de red casi nunca están comprometidos a tal grado. (La gente que compromete a los proveedores de red tiende a trabajar sobre cantidad. Analizar los paquetes les dará muchos servidores con rapidez, pero los ataques de hombre en el camino les dan sólo un sitio a la vez.) Estos ataques tienden a ser problemas sólo si uno de los sitios involucrados tiene usuarios hostiles con acceso físico a la red (por ejemplo, este puede ser el caso si el sitio es una universidad).

Así que ¿en quién puede confiar? Siendo extremistas, en nadie, a menos que confíe en las máquinas involucradas en ambos extremos y la ruta entre ellas. Si confía en las máquinas pero no en la ruta de acceso, puede usar la encriptación para darle la conexión segura sobre una ruta de acceso poco segura. Desafortunadamente, como lo explicamos en el capítulo 10, no hay herramientas ampliamente difundidas y disponibles que hagan eso aún, pero un gran número de sitios está experimentando con soluciones apropiadas y comienzan a aparecer soluciones comerciales.

3.2.20 FILTRADO POR SERVICIO

Como explicamos previamente, bloquear paquetes falsificados de entrada es casi el único uso común de los filtros basados únicamente en direcciones. Muchos otros usos del filtrado de paquetes implican el filtrado por servicio, lo que es un poco más complicado.

Desde el punto de vista del filtrado de paquetes, ¿cómo lucen los paquetes asociados con servicios específicos? Como ejemplo, vamos a observar un Telnet detalladamente. Telnet permite que un usuario inicie sesión en otro sistema, como si tuviera una terminal conectada directamente a él. Usamos Telnet como ejemplo porque es muy común, bastante simple y, desde el punto de vista del filtrado de paquetes, representativo de muchos otros protocolos, como SMTP y NNTP. Debemos observar el servicio Telnet saliente y entrante.

3.2.21 SERVICIO TELNET DE SALIDA

Veamos primero el servicio Telnet de salida, en el cual un cliente local (un usuario)conversa con un servidor remoto. Debemos manejar tanto paquetes de salida como de entrada.

Los paquetes de salida de este servicio contienen los golpes del teclado del usuario y tienen las siguientes características:

- La dirección IP fuente de los paquetes salientes es la dirección IP del anfitrión local
- La dirección IP destino es la dirección IP del anfitrión remoto
- Telnet es un servicio basado en TCP, así que el tipo de paquete IP es TCP
- El puerto destino es el 23; ese es el número de puerto bien conocido que utilizan los servidores Telnet
- El número de puerto TCP fuente (que llamamos "Y" en este ejemplo) es algún número aparentemente al azar mayor a 1023
- El primer paquete salida, que establece la conexión, no tendrá encendido el bit ACK; el resto de los paquetes de salida lo tendrán

Los paquetes entrantes de este servicio contienen la información que aparecerá en la pantalla del usuario (por ejemplo, el indicador login:) y tienen las siguientes características:

- La dirección IP fuente de los paquetes de entrada es la dirección IP del anfitrión remoto
- La dirección IP destino es la dirección IP del anfitrión local
- El tipo de paquete IP es TCP
- El puerto fuente TCP destino es el mismo "Y" que usamos como puerto fuente para los paquetes salientes
- Todos los paquetes entrantes tendrán encendido el bit ACK (de nuevo, sólo el primer paquete, que establece una conexión, no tiene encendido el bit ACK; en este ejemplo, ese primer paquete fue un paquete de salida, no de entrada)

Observe las semejanzas entre los campos de encabezado de los paquetes de salida y de entrada de Telnet. Se usan las mismas direcciones y números de puerto; sólo son intercambiados entre la fuente y el destino. Si compara un paquete saliente con uno entrante, las direcciones fuente y el destino están intercambiadas, al igual que los números de puerto fuente y destino.

¿Por qué está restringido el puerto cliente (el puerto de origen de los paquetes de salida y el puerto destino de los paquetes de entrada) a ser mayor que 1023? Este es un legado de las versiones BSD de UNIX, a las que casi todos los códigos UNIX de red remiten sus orígenes. UNIX BSD reservó de los puertos 0 al 1023 para uso local sólo de root. Estos puertos se utilizan en forma normal sólo por servidores, no por clientes. (Las excepciones principales son los comandos "r" de BSD, tales como rcp y rlogin, como explicamos en el capitulo 8. Otros sistemas operativos, aun los que no tienen conceptos parecidos al de un usuario de red privilegiado (root), por ejemplo, sistemas Macintosh y MS-DOS, han mantenido esta convención. Cuando los programas cliente necesitan un número de puerto para su uso propio, cualquier viejo número de puerto funcionará, los programas serán asignados a un puerto arriba de 1023.

3.2.22 SERVICIO TELNET DE ENTRADA

A continuación, observaremos un servicio de Telnet de entrada, en el cual un cliente remoto (un usuario remeto) se comunica con un servidor Telnet local. De nuevo, debemos manejar tanto paquetes entrantes como salientes.

Los paquetes entrantes para el servicio Telnet contienen los golpes del teclado del usuario, y tienen las siguientes características:

- La dirección IP fuente de estos paquetes es la dirección del anfitrión remoto
- La dirección IP destino es la dirección del anfitrión local
- El tipo de paquete IP es TCP
- El puerto TCP fuente es algún número de puerto al azar mayor a 1023 (el cual llamamos "Z" en este ejemplo)
- El puerto TCP destino es 23
- El bit ACK de TCP no estará encendido en el primer paquete de entrada, que establece la conexión, pero lo estará en todos los paquetes entrantes

Los paquetes de salida de este servicio Telnet contienen las respuestas del servidor (la información que aparecerá en la pantalla del usuario) y tienen las siguientes características:

- La dirección IP fuente es la dirección del anfitrión local
- La dirección IP destino es la dirección del anfitrión remoto
- El tipo de paquete IP es TCP
- El puerto TCP destino es el mismo puerto al azar "Z" que se usó como puerto fuente para los paquetes de entrada
- El bit ACK de TCP estará encendido en todos los paquetes de salida

De nuevo, observe las similitudes entre los encabezados relevantes de los paquetes de entrada y de salida: las direcciones fuente y destino están intercambiadas, lo mismo que los puertos fuente y destino.

3.2.23 RESUMEN DE TELNET

La siguiente tabla ilustra los diversos tipos de paquetes involucrados en servicios Telnet de entrada y salida.

Dirección del servicio	Dirección del paquete	Dirección fuente	Dirección destino	Tipo de paquete	Puerto fuente	Puerto destino	ACK encendido
Salida	de salida	Interna	Externa	TCP	Y	23	a
Salida	de entrada	Externa	Interna	TCP	23	Y	Sí
Entrada	de entrada	Externa	Interna	TCP	Z	23	a
Entrada	de salida	Interna	Externa	TCP	23	Z	Sí

El bit ACK de TCP está encendido en todos los paquetes, excepto el primero, que establece la conexión.

Observe que tanto Y como Z son números de puerto al azar (desde el punto de vista del sistema para filtrado de paquetes)arriba del 1023.

Si quiere permitir Telnet de salida, nada más debe establecer su filtrado de paquetes así:

Regla	Dirección	Dirección fuente	Dirección destino	Protocolo	Puerto fuente	Puerto destino	ACK encendido	Acción
A	Fuera	Cualquiera	Cualquiera	TCP	>1023	23	Cualquiera	Permitir
B	Dentro	Interna	Interna	TCP	23	>1023	Sí	Permitir
C	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Prohibir

- La regla A permite que salgan paquetes hacia servidores Telnet remotos
- La regla B permite la entrada a los paquetes que regresan. Como verifica que el bit ACK este encendido, la regla B no puede utilizarla un atacante para permitir conexiones TCP de entrada de puerto 23 de su extremo a puertos arriba de 1023 en el extremo de usted, por ejemplo, un servidor X11 en el puerto 6000
- La regla C es la regla predeterminada. Si no se aplica ninguna de las reglas anteriores, el paquete se bloquea. Recuerde nuestra explicación: Cualquier paquete bloqueado debe ser registrado y esto puede o no ocasionar que envíe un mensaje de ICMP al emisor

3.2.24 RIESGOS DE FILTRAR POR PUERTO FUENTE

No deja de ser arriesgado tomar decisiones para filtrado basadas en el puerto fuente. Hay un problema fundamental con esta clase filtros: Solo puede confiar en el puerto fuente tanto como confía en la máquina fuerte.

Suponga que por error da por hecho que el puerto fuente esta asociado con un servicio específico. Alguien que tenga el control de la máquina fuente, por ejemplo, alguien con acceso a root en un sistema UNIX (o quien sea que tenga una computadora personal en la red) podría ejecutar el cliente o servidor que desee en un "puerto fuente" que usted permita a través de su cuidadosamente configurado sistema para filtrado de paquetes. Más aún, como explicamos antes, no siempre puede confiar en que la dirección fuente le diga con certeza cual es la máquina fuente; no puede asegurar si esta conversando con la máquina verdadera con esa dirección, o con un atacante que finge ser una máquina.

¿Qué puede hacer al respecto? Debe restringir los números de puertos locales lo máximo posible, sin importar a qué tan pocos puertos remotos les permite el acceso. Si sólo permite conexiones entrantes al puerto 23, y si éste tiene un servidor Telnet digno de confianza (un servidor que sólo hará cosas que un cliente Telnet debe poder decirle que haga), no importa en realidad si el programa con el que está conversando es un cliente Telnet genuino o no. Su preocupación es limitar las conexiones entrantes a sólo los puertos donde ejecuta servidores confiables y asegurarse de que sus servidores son realmente dignos de confianza.

Como muchos servicios usan puertos al azar arriba del 1023 para sus clientes, y puesto que algunos servicios usan puertos arriba del 1023 para los servidores, con frecuencia deberá aceptar paquetes de entrada para puertos que tengan servidores que no sean dignos de confianza. En TCP, puede aceptar paquetes entrantes sin aceptar conexiones de entrada estableciendo el bit ACK.

Cómo elegir un router con filtrado de paquetes.

Hay disponibles varios routers con filtrado de paquetes, algunos buenos y otros tantos no tanto. Casi cada router dedicado soporta el filtrado de paquetes de algún modo. Además están disponibles programas para el filtrado de paquetes para muchos propósitos generales en UNIX y plataformas de computadoras personales.

Debe tener un funcionamiento de filtrado de paquetes suficientemente bueno para sus necesidades.

Mucha gente se preocupa sin necesidad por el funcionamiento del filtrado de paquetes. De hecho, en la mayoría de los Firewalls de Internet, el factor que limita el funcionamiento es la rapidez de la conexión con esa red, no la rapidez del sistema de filtrado.

Por lo general, las conexiones a Internet tiene líneas de 56 Kb/s o de 1.544 Mb/s (T1). El filtrado de paquetes es una operación por paquete. De ahí que mientras menores sean los paquetes, se manejará una mayor cantidad de ellos por segundo y el sistema para filtrado deberá tomar más decisiones cada segundo. El menor tamaño posible de un paquete IP (un paquete simple que contenga sólo un encabezado IP y nada de información) es de 20 Kbytes (160 bits) de longitud. Por lo tanto, una línea de capacidad de 56 Kb/s puede transferir, a (160 bits) de longitud. Por lo tanto, una línea de capacidad de 56 Kb/s puede transferir, a lo sumo, 350 paquetes por segundo,; una línea de 1.544 Mb/s (una T!, por ejemplo) puede transferir, a lo sumo, 9560 paquetes por segundo.

Sin embargo, de hecho verá pocos paquetes IP simples, siempre hay algo en el segmento de información (por ejemplo, un paquete TCP, UDP, o ICMP). Un paquete típico que cruza un Firewall sería un paquete TCP/IP, ya que la mayoría de los servicios de Internet están basados en TCP.

La velocidad es más que un detalle en un Firewall interno en la red de una organización. Tal Firewall necesitará ejecutarse en velocidades LAN, las cuales, en teoría, son por lo común al menos de 10 Mb/s y pueden ser muchos mayores.

Un Firewall con más de dos conexiones también puede tener una velocidad mucho mayor. Con dos conexiones, la velocidad máxima requerida es la de conexión más lenta.

Debe permitir reglas basadas en cualquier encabezado o criterio de metapaquete.

Debe ser capaz de especificar reglas basadas en cualquier información del encabezado o de metapaquete disponible para sus paquetes. La información del encabezado incluye lo siguiente:

* Dirección IP fuente y destino.
* Opciones IP.
* Protocolo, como TCP, UDP o ICMP.
* Puerto TCP o UDP fuente y destino.
* Tipo de mensaje de ICMP.
* Información de inicio de conexión (bit ACK) para paquetes TCP

e información similar para cualquier otro protocolo sobre el que esté haciendo filtrado. La información de metapaquete incluye cualquier dato sobre el paquete conocido por el router pero que no está en los propios encabezados; por ejemplo, por cual interface del router entró o por cual va a salir. Debe especificar reglas basadas en las combinaciones de estos criterios de encabezados y metapaquetes.

Las reglas deben aplicarse en el orden especificado

Su filtro de paquetes debe aplicar, en un orden predecible, las reglas que le especificó. La orden más simple es aquella en la que usted, la persona que configura el router, especifica las reglas. Desafortunadamente, algunos productos, en lugar de aplicar las reglas en el orden en que se especificó, intentan reordenarlas y combinarlas para alcanzar mayor eficiencia al aplicarlas. Esto ocasiona varios problemas:

* Al reordenarse las reglas, se le dificulta a usted saber lo que pasa y lo que hará el router con un conjunto específico de instrucciones de filtrado. Configurar un sistema para filtrado de paquetes es ya de por sí bastante complicado, aun antes de que un proveedor agregue complicaciones al fusionar y reordenar los grupos de reglas.

* 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 reordenar las reglas puede deshacerse un conjunto que sólo trabajaría bien si no hubiera sido reordenado.

Consideremos un ejemplo. Imagine que trabaja en una corporación en un proyecto especial con una universidad local. Su red corporativa. Su red corporativa clase B es la 172.16 (esto es, sus direcciones IP van de la 172.16.0.0 a la 172.16.255.255). La universidad tiene la red 10 clase A (esto es, sus direcciones IP van de la 10.0.0.0 a la 10.255.255.255).

Para los fines de ese proyecto, se unió su red directamente a la de la universidad, usando un router con filtrado de paquetes. Quiere deshabilitar todo el acceso a Internet a través de ese enlace (el acceso a Internet debe ser a través de Firewall para Internet). Su proyecto especial con la universidad usa la subred 172.16.6 de su red clase B (esto es, direcciones IP de la 172.16.6.0 a la 172.16.6.255). Quiere que todas las subredes de la universidad puedan tener acceso a la subred de este proyecto. En la universidad hay una subred de ocho bits, la red 10.1.99 la cual tiene mucha actividad hostil. Debe asegurarse de que esta subred sólo llegue a la subred de su proyecto.

¿Cómo puede conjuntar todos estos requisitos? Especifique las tres reglas para filtrado de paquetes que se muestran abajo.

Regla	Dirección fuente	Dirección destino	Acción
A	10.0.0.0/8	172.16.6.0/24	Permitir
B	10.1.99.0/24	172.16.0.0/16	Prohibir
C	Cualquiera	Cualquiera	Prohibir

* La reglaA permite que la universidad tenga acceso a la subred de su proyecto

* La regla B bloquea a la red hostil de la universidad al resto de su red

* la regla C deshabilita el acceso a Internet a su red.

Veamos ahora lo que ocurre en diferentes casos, dependiendo de cómo se apliquen estas reglas.

Si las reglas se aplican en el orden ABC

Si las reglas se aplican en el orden ABC (el mismo orden especificado por el usuario), la siguiente tablas muestra lo que ocurre con una variedad de paquetes de ejemplo.

Paquete	Dirección fuente	Dirección destino	Acción deseada	Acción real (por regla)
1	10.1.99.1	172.16.1.1	Prohibir	Prohibir (B)
2	10.1.99.1	172.16.1.1	Permitir	Permitir (A)
3	10.1.1.1	172.16.1.1	Permitir	Permitir (A)
4	10.1.1.1	172.16.1.1	Prohibir	Prohibir (C)
5	192.168.3.4	172.16.1.1	Prohibir	Prohibir (C)
6	192.168.3.4	172.16.1.1	Prohibir	Prohibir (C)

* El paquete 1 va de una máquina en la universidad que está en la subred hostil hacia una máquina cualquiera en su red (no en la subred del proyecto) quiere prohibir esto,; lo está, por regla B.

* El paquete 2 va de una máquina en la universidad que está en la subred hostil hacia una máquina en la subred de su proyecto; quiere que eso éste permitido; lo está, por regla A.

* El paquete 3 va de una máquina cualquiera en la universidad hacia una máquina en la subred de su proyecto; quiere que eso esté permitido; lo está, por regla A.

* El paquete 4 va de una máquina cualquiera en la universidad a una de sus máquinas no incluidas en el proyecto; quiere prohibir esto; lo está, por la regla C.

* El paquete 5 va de una máquina cualquiera de Internet hacia una de sus máquinas que no son del proyecto; quiere prohibir esto; lo está, por la regla.

* El paquete 6 va de una máquina cualquiera de Internet a una de sus máquinas del proyecto; quiere que esto esté negado; lo está, por la regla C.

Por lo tanto, si se aplican las reglas en el orden ABC, ocurre lo que usted quiere.

Si las reglas se aplican en el orden BAC

¿Qué pasaría si el router volviera a ordenar las reglas por el número de bits significativos en la dirección fuente, para que las reglas más específicas se apliquen primero? En otras palabras, ¿qué pasaría si las reglas que se aplican a direcciones IP fuente más específicas (esto es, reglas que se aplican a un menor rango de direcciones fuente) se aplican antes que las que se aplican a direcciones IP fuente menos específicas? En este caso, las reglas se aplicarían en el orden BAC.

Regla	Dirección fuente	Dirección destino	Acción
B	10.1.99.0/24	172.16.0.0/16	Prohibir
A	10.0.0.0/8	172.16.6.0/24	Permitir
C	Cualquiera	Cualquiera	Prohibir

Aquí aparecen los mismos seis paquetes del ejemplo, con la variante de que las reglas se aplican en el orden BAC:

Paquete	Dirección fuente	Dirección destino	Acción deseada	Acción real (por regla)
1	10.1.99.1	172.16.1.1	Prohibir	Prohibir (B)
2	10.1.99.1	172.16.1.1	Permitir	**Prohibir (B)**
3	10.1.1.1	172.16.1.1	Permitir	Permitir (A)
4	10.1.1.1	172.16.1.1	Prohibir	Prohibir (C)
5	192.168.3.4	172.16.1.1	Prohibir	Prohibir (C)
6	192.168.3.4	172.16.1.1	Prohibir	Prohibir (C)

Si se aplican las reglas en el orden BAC, entonces el paquete 2, que debería permitirse, será prohibido incorrectamente por la regla B. Ahora, prohibir algo que debía permitirse es más seguro que permitir algo que debería prohibirse, pero sería mejor si el sistema para filtrado se limitara a hacer lo que usted quería que hiciera.

La regla B no es realmente necesaria

Si analiza este ejemplo con cuidado, puede ver que la explicación sobre la subred hostil, que es la razón de existir de la regla B, es redundante y no es necesaria para obtener el resultado deseado. La regla B está destinada a limitar el acceso de la subred hostil a únicamente la subred de su proyecto. Sin embargo, la regla A ya restringe a toda la universidad, incluyendo la subred hostil, el acceso sólo a la subred de su proyecto. Si omite la regla B, entonces las reglas se aplicarán en el orden AC sin importar si el sistema las vuelve a ordenar basado en el número de bits significativos en la dirección IP fuente. Las tablas de abajo muestran qué ocurre en cada caso:

Regla	Dirección fuente	Dirección destino	Acción
A	10.0.0.0/8	172.16.6.0/24	Permitir
C	Cualquiera	Cualquiera	Prohibir

Paquete	Dirección fuente	Dirección destino	Acción deseada	Acción real (por regla)
1	10.1.99.1	172.16.1.1	Prohibir	Prohibir (C)
2	10.1.99.1	172.16.1.1	Permitir	Permitir (A)
3	10.1.1.1	172.16.1.1	Permitir	Permitir (A)
4	10.1.1.1	172.16.1.1	Prohibir	Prohibir (C)
5	192.168.3.4	172.16.1.1	Prohibir	Prohibir (C)
6	192.168.3.4	172.16.1.1	Prohibir	Prohibir (C)

Las reglas para filtrado de paquetes son engañosas.

El punto aquí es arreglar bien las reglas para filtrado es engañoso. En este ejemplo, analizamos una situación simple y aun así creamos un conjunto de reglas que tenía un sutil error. Los conjuntos de reglas de la vida real son muchos más complejos y con frecuencia incluyen decenas o centenares de reglas. Es casi imposible considerar las implicaciones e interacciones de todas esas reglas, a menos que simplemente se apliquen en el orden especificado.

Debe ser capaz de registrar los paquetes acotados y los desechados

Asegúrese de que su router con filtrado de paquetes le dé la opción de registrar todos los paquetes que desecha. Quiere saber sobre cualquier paquete que sea bloqueado por sus reglas de filtrado. Estas reglas reflejan sus políticas de seguridad y usted quiere saber cuando alguien intenta violarlas. El modo más sencillo de aprender de esos intentos de violación es a través de un registro.

También le gustaría ser capaz de registrar los paquetes que fueron aceptados. por ejemplo, tal vez quiera registrar el inicio de cada conexión TCP. Registrar todos los paquetes aceptados es demasiada información en una operación normal, pero puede valer la pena de manera ocasional para depurar el sistema y afrontar los ataques en potencia. Aunque probablemente hace algún registro en el destino del paquete, ese registro no funciona si los hosts destino están comprometidos y no muestran los paquetes que lograron pasar a través del filtro de paquetes aunque no tienen un destino válido. Esos paquetes son interesantes, ya que pueden ser pruebas de que hay un atacante. Sin información el router, no tendrá una visión completa de lo que hacen los atacantes.

Ejemplo del filtrado de paquetes.

Permitir SMTP de entrada y salida (para que pueda enviar y recibir correo electrónico), nada más. Puede empezar el siguiente conjunto de reglas.

* Las reglas A y B permiten conexiones SMTP de entrada (correo electrónico de entrada)

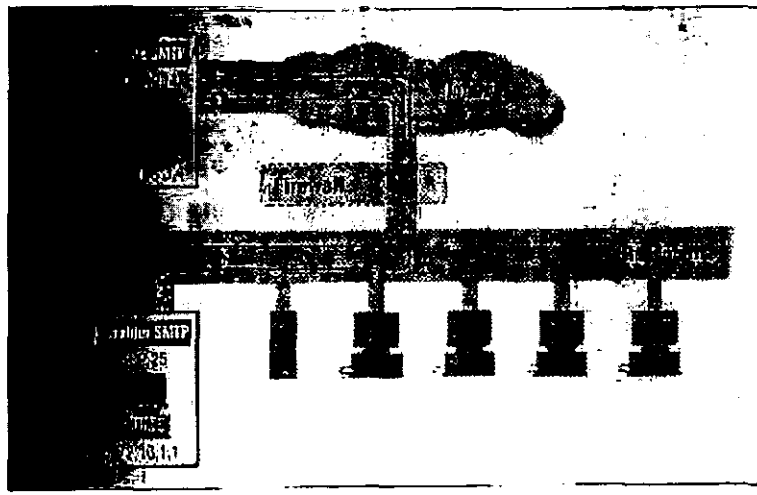
* Las reglas C y D permiten conexiones SMTP de salida (correo electrónico de salida).

* La regla E es la predeterminada, que se aplica si todo lo demás falla.

Ahora, pongamos algunos paquetes de ejemplo para ver qué ocurre. Digamos que su host tiene la dirección IP 172.16.1.1 y que alguien intenta enviarle correo de un host remoto con dirección IP 192.168.3.4. Además, digamos que el cliente SMTP emisor usa el puerto 1234 para conversar con su servidor SMTP, que está en el puerto 25

Paquete	Dirección	Dirección fuente	Dirección destino	Protocolo	Puerto destino	Acción
1	Dentro	192.168.3.4	172.16.1.1	TCP	25	Permitir(A)
2	Fuera	172.16.1.1	192.168.3.4	TCP	1234	Permitir(B)

La fig. muestra este caso.



En este caso, las reglas para filtrado de paquetes le permiten la entrada al correo electrónico:

* La regla A permite los paquetes entrantes del cliente SMTP que envía a su servidor SMTP (arriba como el paquete 1)

* Las regla B permiten que regresen las respuestas de su servidor hacia el cliente emisor (arriba como el paquete 2)

¿Qué ocurre con su correo electrónico de salida hacia ellos? Digamos que su cliente SMTP usa el puerto 1357 para conversar con el servidor SMTP de ellos:

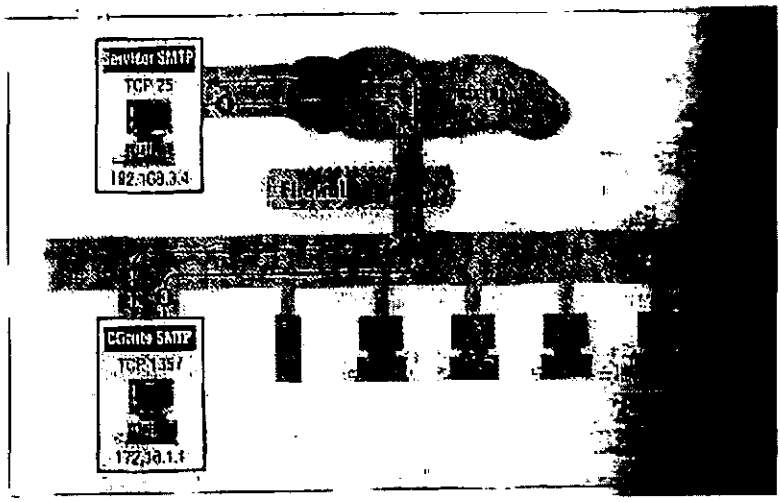
Paquete	Dirección	Dirección fuente	Dirección destino	Protocolo	Puerto destino	Acción
3	Fuera	172.16.1.1	192.168.3.4	TCP	25	Permitir(C)
4	Dentro	192.168.3.4	172.16.1.1	TCP	1357	Permitir(D)

De nuevo, aquí las reglas para el filtrado de paquetes permiten la salida de su correo electrónico:

- La regla C permite paquetes de salida de su cliente SMTP al servidor SMTP de ellos (arriba por el paquete 3)
- La regla D permite que regresen las respuestas del servidor de ellos al cliente de usted (arriba por el paquete 4)

Ahora cambiemos las coas. ¿Qué pasa si alguien del mundo exterior (por ejemplo, alguien en el host 10.1.2.3) intenta abrir una conexión desde el puerto 5150 al servidor x11 del puerto 6000 en uno de los sistemas externos de usted (por ejemplo 172.16.3.4) para llevar a cabo un ataque?

Paquete	Dirección	Dirección fuente	Dirección destino	Protocolo	Puerto destino	Acción
5	Dentro	10.1.2.3	172.16.3.4	TCP	25	Permitir(C)
6	Fuera	172.16.3.4	10.1.2.3	TCP	1357	Permitir(D)



El conjunto de reglas mostrado arriba permite que esta conexión tenga éxito! De hecho, el conjunto de reglas mostrado arriba permite que cualquier conexión tenga lugar mientras ambos extremos utilicen puertos arriba del 1023. ¿Por qué?

- Las reglas A y B juntas hacen que usted permita las conexiones SMTP de entrada.
- Las reglas C y B juntas hacen que usted permita las conexiones SMTP de salida.

- Pero las reglas B y D juntas terminan por permitir todas las conexiones donde ambos extremos utilicen puertos arriba del 1023 y, por supuesto, esto no es lo que intentaba hacer.

Quizá hay muchos servidores vulnerables ateniendo ("escuchando") en los puertos de arriba del 1023 en su sitio. Ejemplos son X11 (puerto 6000), base de datos (Sybase, Oracle, Informix y otras bases utilizan puertos elegidos arriba del 1023) y muchos otros.

3.3 SISTEMAS PROXY

Un sistema proxy proporciona acceso a Internet a un sólo hosts, o a un número muy pequeño de servidores, aunque parece que lo proporciona a todos. Los servidores que sí tienen acceso actúan como proxys para las máquinas que no lo tienen, haciendo lo que estás últimas quieren que haga.

Un servidor proxy para un protocolo o conjunto de protocolos determinados se ejecuta en un servidor con doble acceso o en un host bastión: algún host con el que pueda establecer comunicación con el usuario, que puede, a su vez, comunicarse con el mundo exterior. El programa cliente del usuario se comunica ("habla") con este servidor proxy en lugar de hacerlo directamente con el servidor "real" que está en el Internet.

El servidor proxy evalúa las solicitudes del cliente y decide cuáles pasar y cuáles no. Si una petición es aprobada, el servidor proxy habla con el servidor real en nombre del cliente (de ahí el término proxy, que significa representante en español) y procede a transmitir las solicitudes del cliente al verdadero servidor y a transmitir las respuestas de éste de nuevo al cliente.

Para el usuario, hablar con el servidor proxy es como hablar directamente con el verdadero servidor. En cuanto a este último, habla con un usuario en el servidor que ejecuta el servidor proxy; no sabe que el usuario en realidad está en otro lugar.

Un sistema proxy no requiere de hardware especial, aunque sí de un software especial para la mayoría de los servicios.

NOTA

Los sistemas proxy son efectivos sólo cuando se utilizan junto con algún método para restringir el tráfico a nivel de IP entre los clientes y los verdaderos servidores, como un router de protección o un host con doble acceso que no enruta paquetes. Si hay conexión a nivel de IP entre los clientes y los verdaderos servidores, los clientes pueden saltarse el sistema proxy.

3.3.1 POR QUÉ UTILIZAR UN PROXY

No tiene el objeto de conectarse a Internet si los usuarios no pueden tener acceso a ella. Por otro lado, no hay seguridad alguna al conectarse a Internet si hay libre acceso entre ella y cada servidor en su sitio. Debe aplicar algún tipo de control.

El control más obvio es proporcionar un sólo servidor que tenga acceso a Internet para todos los usuarios. Sin embargo, esta no es una solución satisfactoria porque estos servidores no son transparentes para los usuarios. Si éstos quieren tener acceso a los servicios de red no pueden hacerlo de manera directa. Deben iniciar una sesión co el servidor con doble acceso, hacer todo su trabajo desde ahí y luego, de alguna forma, transferir el resultado de nuevo a sus estaciones de trabajo. En el mejor de los casos, este proceso de múltiples pasos resulta molesto para los usuarios, pues los obliga a realizar múltiples transferencias y trabajar sin personalización a la que están acostumbrados.

Los sistemas proxy evitan la frustración del usuario y las inseguridades de un host con doble acceso. Manejan aquélla automatizando la interacción con tal servidor. En lugar de requerir que los usuarios traten en directo con el host de doble acceso, los sistemas proxy permiten que toda la interacción se lleve a cabo atrás del escenario. El usuario tiene la ilusión de que trata directamente (o casi de forma directa) con el servidor que está en Internet y con el que en realidad desea establecer contacto, con un mínimo de interacción directa con el host de doble acceso.

Los sistemas proxy manejan los problemas de inseguridad evitando que los usuarios inicien sesiones con el host de doble acceso y forzando las conexiones a través de software controlado. Debido a que el software proxy funciona sin que los usuarios inicien una sesión. También es imposible para alguien instalar software que no está controlado para tener acceso a Internet; el proxy actúa como punto de control.

3.3.2. VENTAJAS DEL USO DE PROXY

Existen varias ventajas en el uso de servicios proxy.

Los servicios proxy permiten a los usuarios acceder en forma "directa" a Internet

Con el método de host de doble acceso, un usuario debe iniciar una sesión con el servidor antes de utilizar cualquier servicio a Internet, lo cual, con frecuencia, es inconveniente, pues algunos usuarios se sienten tan frustrados que buscan formas de evadir el Firewall. Con los servicios proxy los usuarios creen que están interactuando en forma directa con los servicios de Internet.

Por supuesto, hay mucho más tras bambalinas, pero casi siempre es transparente para los usuarios. Aunque los servicios proxy permiten a los usuarios a los servicios de Internet desde sus propios sistemas, lo hacen sin permitir que los paquetes pasen directamente entre el sistema del usuario e Internet. La ruta indirecta, a través de un host con doble acceso, o mediante una combinación de host bastión y un router de protección.

Los servicios proxy son buenos para la contabilidad del sistema

Debido a que los servidores proxy comprenden el protocolo de niveles inferiores, permiten que se lleve la contabilidad del sistema de una forma muy efectiva. Por ejemplo, en lugar de registrar todos los datos transferidos, un servidor proxy FTP registra sólo los comandos emitidos y las respuestas recibidas del servidor, lo cual resulta en un registro mucho más pequeño y útil.

3.3.3 DESVENTAJAS DEL USO DE PROXY

También existen algunas desventajas en el uso de servicios proxy.

Los servicios proxy son más lentos que los servicios no proxy

Aunque los programas proxy están ampliamente disponibles para los servicios más viejos y simple, como FTP y Telnet, es más difícil encontrar software para servicios nuevos o poco comunes. Es usual que exista un rezago entre la introducción de un servicio y la disponibilidad de los servidores proxy para él; el rezago depende, sobre todo, de cuán bien esté diseñado el servicio para utilizarse en un proxy. Esto dificulta que un sitio ofrezca nuevos servicios inmediatamente después de que estén disponibles. Hasta que no se disponga de un software proxy adecuado, un sistema que necesite nuevos servicios tal vez deba ponerse fuera de un Firewall, lo cual abre agujeros potenciales en la seguridad.

Los servicios proxy podrían requerir servidores diferentes para cada servicio

Quizá necesita un servidor proxy para cada protocolo, porque el servidor proxy debe comprender el protocolo para determinar qué permite y qué no permite, y para hacerse pasar como cliente ante el verdadero servidor y como el verdadero servidor ante el cliente proxy. Coleccionar, instalar y configurar todos estos servidores puede requerir mucho trabajo.

Los productos y la paquetería difieren mucho en cuanto a la facilidad con la que pueden ser configurados, pero facilitar las cosas en un lugar puede hacerlas más difíciles en otros. Por ejemplo, los servidores fáciles de configurar por lo general están limitados en flexibilidad; son fáciles de configurar porque hacen ciertas suposiciones sobre cómo piensan que van a ser utilizados, que puede o no ser lo correcto o apropiado para su sitio.

Los servicios proxy por lo general requieren de modificaciones a los clientes, a los procedimientos o a ambos

Excepto por contados servicios diseñados para emplearse con proxy, los servidores proxy necesitan modificaciones a los clientes y/o procedimientos. Cualquier tipo de modificación tiene desventajas, las personas no siempre pueden utilizar las herramientas que están fácilmente disponibles con sus instrucciones normales.

Debido a estas modificaciones, las aplicaciones proxy no funcionan tan bien como aplicaciones no proxy. Tienden a modificar las especificaciones de los protocolos, y algunos clientes y servidores son menos flexibles que otros.

Los servicios proxy no lo protegen de todas las debilidades de los protocolos.

Como solución para la seguridad, el uso de proxy depende de la habilidad para determinar qué operaciones son seguras en un protocolo. No todos los protocolos proporcionan formas fáciles de hacer esto. HTTP está diseñado para operar eficazmente con servidores proxy, pero también está diseñado para que pueda extenderse con facilidad, y logra esa meta pasando datos que van a ejecutarse. Es imposible que un servidor proxy lo proteja de los datos; tendría que entender los datos que pasan y determinar si son peligrosos o no.

3.3.4 CÓMO FUNCIONA UN PROXY.

Los detalles de cómo funciona un proxy difieren de un servicio a otro. Algunos servicios proporcionan un proxy fácil o automáticamente; para esos servicios, un proxy se instala al realizar cambios de configuración a los servidores normales. Sin embargo, para la mayoría de los servicios la instalación de un proxy requiere de software apropiado para un servidor proxy del lado del servidor. Del lado del cliente, necesita uno de los siguientes:

Software cliente personalizado

Con este enfoque, el software debe saber cómo ponerse en contacto con el servidor proxy en lugar de con el verdadero servidor cuando un usuario hace una solicitud (por ejemplo, para FTP o Telnet), y cómo decirle al servidor proxy con cuál servidor real conectarse.

Procedimientos personalizados de usuario.

Con este enfoque, el usuario utiliza software cliente estándar para hablar con el servidor proxy y le dice que se conecte con el verdadero servidor, en lugar de conectarse con el verdadero servidor directamente.

Uso de software cliente personalizado para proxy.

El primer enfoque es utilizar software cliente personalizado para proxy. Hay algunos problemas asociados con este enfoque.

Con frecuencia, el software cliente personalizado adecuado está disponible sólo para ciertas plataformas. Si no se halla disponible para una de sus plataformas, sus usuarios no corrieron con suerte. Por ejemplo, el programa Igateway de Sun es un paquete proxy para FTP y Telnet, pero sólo puede utilizarlo en máquinas Sun porque proporciona sólo binarios precompilados de Sun. Si va a utilizar software proxy, es obvio que debe seleccionar el que esté disponible para las plataformas necesarias.

Aunque el software esté disponible para sus plataformas, quizá no sea el que quieren sus usuarios. Por ejemplo, en Mcintosh hay docenas de programas FTP cliente. Algunos tienen interface gráficas de usuario realmente impresionantes; otros tienen otras características útiles, por ejemplo, anarchie es un programa que combina un cliente Archie y un cliente FTP en un programa. En algunos casos, tal vez pueda modificar clientes para soportar su servidor proxy, pero hacerlo requiere de que tenga el código fuente para el cliente, así como las herramientas y la habilidad de recompilado. Pocos programas cliente vienen con soporte para algún tipo de proxy.

La feliz excepción a esta regla son los programas cliente para el WWW, como Mosaic. Muchos de estos programas soportan varios tipos. La mayoría de estos programas son bastante nuevos y, por lo tanto, fueron escritos después de que los Firewalls y los sistemas proxy se habían vuelto comunes en Internet; reconociendo el ambiente en el que estarían trabajando, sus autores optaron por un diseño que soportara proxy desde el principio.

El uso de clientes modificados para proxy no hace que éste sea completamente transparente para los usuarios. La mayoría de los sitios utilizan clientes sin cambios para conexiones internas y los modificados sólo para realizar conexiones externas. Seguir los

procedimientos que están acostumbrados a utilizar en otra parte, o los que están escritos en los libros, puede dejarlos perplejos ante los resultados aparentemente intermitentes cuando las conexiones internas tienen éxito y las externas fallan. (El uso interno de los clientes funcionará, pero introduce dependencias innecesarias en el servidor proxy; esa es la razón por la cual la mayoría de los sitios evitan).

Uso de procedimientos personalizados de usuarios proxy

Con el enfoque de procedimiento personalizado, los servidores proxy están diseñados para trabajar con programas estándar; sin embargo, requieren que los usuarios de los programas sigan procedimientos personalizados. El usuario le dice al cliente que se conecte con el servidor proxy y luego le dice a este último a qué servidor conectarse. Puesto que pocos protocolos están diseñados para pasar este tipo de información, el usuario debe recordar no sólo el nombre del servidor proxy sino, también, qué medios especiales se emplean para pasar el nombre del otro servidor.

¿ Cómo funciona esto? Usted debe enseñar a sus usuarios procedimientos específicos que deben seguir para cada protocolo. Veamos FTP. Supongamos que un usuario quiere obtener un archivo de un servidor FTP anónimo. He aquí lo que hace el usuario:

1. Usando cualquier cliente FTP, se conecta con un servidor proxy (el cual es probable que esté ejecutándose en el host bastión, la compuerta a Internet) en lugar de al servidor FTP anónimo.

2. En el indicador del sistema, además de especificar el nombre que quiere utilizar, el usuario también indica el nombre que quiere utilizar, el usuario también indica el nombre del verdadero servidor al que quiere conectarse.

Así como el uso del software personalizado requiere cierta modificación de los procedimientos del usuario, el uso de procedimientos personalizados pone limitaciones sobre los clientes que puede utilizar. Algunos clientes intentan hacer FTP anónimo automáticamente; no sabrán como pasar a través del servidor proxy. Algunos clientes pueden inteferir en formas más simples, por ejemplo, proporcionando una interface gráfica de usuario que no permite que usted escriba un nombre de usuario lo suficientemente largo como para que incluya el nombre de usuario y del el servidor.

El principal problema con el uso de procedimientos personalizados es que tiene que enseñárselos a sus usuarios y , además, están preparados técnicamente, puede o no ser un problema. Sin embargo, si tiene diez mil usuarios esparcidos en cuatro continentes, será un problema.

3.3.5 TERMINOLOGÍA PARA SERVIDORES PROXY.

Proxy a nivel aplicación en comparación de proxy a nivel circuito.

Un proxy a nivel aplicación es el que sabe sobre la aplicación específica para la cual está proporcionando servicios proxy; comprende e interpreta los comandos en el protocolo de la aplicación. Un proxy a nivel circuito es el que crea un circuito entre el cliente y el servidor sin interpretar el protocolo de la aplicación. La versión más extrema de un proxy a nivel de aplicación es una aplicación como Sendmail, que implementa un protocolo de guardar y enviar.

La versión más extrema de un proxy a nivel circuito es una de las modernas compuertas proxy híbridas que parecen como proxy para el exterior pero como router con el filtrado para el interior.

Para poder establecer una conexión proxy, usted debe saber adónde se supone que se dirige ésta, una compuerta híbrida puede simplemente, interceptar conexiones que va hacia él; algo más que debe indicarle es en dónde hacer la conexión hacia afuera. Un proxy a nivel aplicación puede obtener esa información del protocolo de aplicación. Un proxy a nivel de circuito no puede interpretar el protocolo de aplicación y necesita que le proporcionen la información a través de otros medios. Como la habilidad para ocupar clientes sin modificar es una característica útil, en general el uso edl proxy a nivel de aplicación está diseñado para aprovechar sus conocimientos del protocolo de aplicación. Un proxy a nivel circuito por lo general no tiene forma de utilizar los procedimientos modificados, así que usa clientes modificados.

Aunque no se sabe de la existencia del algún proxy a nivel de aplicación con cliente modificado, sí hay proxy a nivel de circuito con procedimiento modificado.

La ventaja de un proxy a nivel e circuito es que proporciona servicios para una amplia gama de protocolos. La mayoría de los servidores proxy a nivel circuito también son servidores proxy genéricos; pueden adaptarse para servir casi a cualquier protocolo. No todos los protocolos pueden manejarse fácilmente por un proxy a nivel circuito. Los protocolos como FTP, que comunican datos del puerto cliente al servidor, necesitan cierta intervención a nivel protocolo y, por lo tanto, ciertos conocimientos a nivel aplicación. La desventaja de un servidor proxy a nivel circuito es que proporciona muy poco control sobre lo que sucede a través del proxy. Al igual que un filtro de paquetes, controla las conexiones con bas en su fuente y destino y no puede determinar muy fácilmente si los comandos que están pasando a través de él son seguros o están en el protocolo esperado. Un proxy a nivel de circuito es fácilmente engañable por servidores instalados en los números de puerto asignados a otros servidores.

Proxy genéricos en comparación con proxy dedicados.

Aunque a nivel "aplicación" y "a nivel circuito" son términos utilizados a menudo, con mayor frecuencia distinguimos entre servidores proxy "dedicados" y "genéricos". Un servidor proxy dedicado es el que sirve a un solo protocolo; un servidor proxy genérico es el que sirve a varios protocolos. En la práctica, los servidores proxy dedicados son a nivel aplicación; los servidores proxy genéricos son a nivel circuito. Dependiendo de cómo considere los matices de significado, podría ser posible producir un servidor proxy genérico a nivel aplicación (que entienda una amplia gama de protocolos) o un servidor proxy dedicado a nivel circuito (que proporcione sólo un servicio pero no entienda el protocolo). Sin embargo, ninguno de éstos existió jamás, así que utilizamos "dedicado" y genérico" sólo porque creemos que son términos más intuitivos que "a nivel aplicación" y "a nivel circuito".

Servidores proxy inteligentes

Un servidor proxy puede hacer mucho más que sólo transmitir peticiones,; el que lo hace es un servidor proxy inteligente. Por ejemplo, el servidor proxy CERN HTTP utiliza el caché de datos; así, las distintas peticiones para los mismos datos no salen a través de Internet. Los servidores proxy (en especial los servidores a nivel aplicación) pueden proporcionar mejores enlaces y controles de acceso que los logrados por medio de otros métodos, aunque poco

servidores proxy existentes aprovechan totalmente las oportunidades. Es más fácil para un servidor proxy dedicado a nivel aplicación ser inteligente; un proxy a nivel circuito tiene habilidades limitadas.

Uso de un proxy con servicios de Internet

Debido a que un proxy interfiere con las comunicaciones entre un cliente y un servidor, debe estar adaptado a cada servicio por separado. Algunas cosas que son fáciles de hacer comúnmente se complican más cuando está involucrado un proxy.

El servicio ideal para un proxy hace una conexión TCP en una dirección; sólo tiene comandos seguros; tiene alguna pieza de datos especificados por el usuario y de la longitud variable que pasan al servidor; y se utiliza desde un cliente interno a un servidor externo.

TCP en comparación con otros protocolos

Debido a que TCP es un protocolo orientado a conexión, usted realiza sólo una vez el gasto de establecer la conexión proxy, y luego sigue utilizándola. UDP no tiene el concepto de conexión; cada paquete es una operación independiente que requiere de una decisión independiente por parte del servidor proxy. TCP es más fácil de utilizar con un proxy (aunque el programa UDP Packet Relayer es un servidor proxy UDP genérico). ICMP es de tan bajo nivel, que es casi imposible usarlo con un proxy.

Conexiones unidireccionales frente a conexiones multidireccionales.

Es fácil para un servidor proxy interceptar la conexión inicial de un cliente a un servidor. Es más difícil para él interceptar una conexión de regreso. Ya sea que ambos extremos de la conversación estén enteradas de la existencia del servidor proxy, o que el servidor sea capaz de interpretar y modificar el protocolo para asegurar que la conexión de regreso esté hecha de manera correcta. Por ejemplo, FTP, normalmente, requiere que el servidor proxy intercepte el comando PORT que envía el cliente al servidor, abra una conexión del proxy al cliente que está en ese puerto, y envíe otro comando PORT (para un puerto en el proxy) al verdadero servidor. No es suficiente con que el servidor proxy simplemente lea el comando PORT conforme pasa, porque ese puerto pueda estar ya en uso.

Cualquier cosa más compleja que una conexión de salida y regreso aún peor.

Seguridad de protocolo.

Para algunos servicios, el uso de proxy puede ser técnicamente fácil, pero inútil desde un punto de vista de seguridad. Si un protocolo es inherentemente inseguro, utilizado por medio de un proxy sin hacer ninguna otra cosa no lo hará más seguro. Por ejemplo, X11 es un poco difícil de emplear con proxy, pero la verdadera razón de que no se use mucho con proxy a través de Firewalls no tiene que ver con los aspectos técnicos (los servidores proxy X no son poco comunes como formas de extender las capacidades de X).

Si es difícil distinguir entre operaciones seguras e inseguras en protocolo, o es imposible de utlizar el servicio si las operaciones inseguras son evitadas, quizá el uso del proxy no sea una solución viable.

Datos especificados por el usuario.

Si va a utilizar un proxy con procedimiento modificado, necesita una parte modificable del procedimiento. Los programas, como los clientes FTP y HTTP, en dónde el cliente le pasa al servidor una cadena larga especificada por el usuario, son perfectos. (Los clientes FTP pasan un nombre de usuario al servidor; los clientes HTTP pasan un URL.) Un programa como ping, en donde los clientes no pasan ningún dato al servidor, es en esencia imposible de usar con proxy si se tiene un cliente sin modificar.

Clientes internos frente a clientes externos

La mayoría de los servidores proxy están diseñados para situaciones en donde el cliente está adentro y el servidor afuera del Firewall. Esto se debe a que la mayoría de los servidores proxy requieren cierta cooperación en el extremo del cliente, y es probable que los clientes modificables y los usuarios que se pueden entrenar estén, ambos, dentro del Firewall. Un proxy para clientes externos funciona sólo en contadas situaciones:

- Empleando procedimientos modificados para proporcionar servicios de entrada a sus propios usuarios.
- Usando algo como plug-gw para redirigir conexiones desde el servidor proxy a una máquina interna. Un programa de este tipo soportará cualquier número de clientes, siempre y cuando todos quieran conectarse al mismo servidor interno.
- Proporcionando un servicio especial para el cual usted distribuye clientes. Si escribe su propio servicio de Internet, peued diseñar con facilidad los clientes para incluir soporte a proxy.

Uso de proxy sin servidor

Algunos servicios, en especial los del tipo "guardar y enviar", como SMTP, NNTP y NTP, por supuesto quem soportan proxy. Todos estos servicios están diseñados para que los mensajes (de correo electrónico para SMTP, de noticias de Usenet para NNTP y los valores de hora para NTP) sean recibidos por un servidor y luego guardados hasta que puedan ser remitidos a otro servidor o servidores apropiados. Para SMTP, los mensajes son enviados hacia un destino de mensajes de correo electrónico. Para NNTP y NTP, los mensajes son remitidos a todos los servidores vecinos. Con un esquema similar, cada servidor intermedio actúa efectivamente como un proxy para el remitente o servidor original.

Si examina los encabezados de "Recibido:" del correo electrónico entrante de Internet (estos encabezados le dan seguimiento a la ruta de un mensaje a través de la red, desde el remitente al receptor), descubre con rapidez que muy pocos mensajes viajan en forma directa desde la máquina del remitente a la del receptor. Es mucho más común en estos tiempos que el mensaje pase a través de cuatro máquinas por lo menos:

- La del remitente
- La compuerta de salida de correo que está en el sitio del remitente
- La compuerta de entrada de correo en el sitio del receptor
- Por último, la máquina del receptor.

Cada uno de los servidores intermedios (las compuertas de correo) actúa como un servidor proxy para el remitente, aunque éste no trate con ellos de manera directa. La fig7.2

3.3.6 USO DE SOCKS PARA PROXY

El programa SOCKS, escrito originalmente por David Koblas Y Michelle Koblas y mantenida actualmente por Ying-Da-Lee, es un ejemplo del tipo de sistema proxy que requiere de clientes personalizados.

SOCKS está disponibles en forma gratuita, y se ha convertido, de hecho, en estándar para proxy de Internet: se ha escrito una solicitud de comentarios (RFC, Request For Comments) y en la actualidad se lleva a cabo el proceso de aprobación.

Para facilitar el soporte de clientes nuevos, SOCKS es excesivamente genérico. Es parte de la razón por la cual es tan popular, pero tiene la desventaja de que no puede proporcionar aceso inteligente o control de acceso. Brinda acceso al sistema, pero la mayor parte se efectúa en el cliente, haciendo difícil reunir la información en un sólo lugar para examinarla. SOCKS sí regsitra las solicitudes de conexión en el servidor; proporciona control de acceso por fuente, y servidor y protocolo destino; y permite respuestas configurables para las negociaciones de acceso. Por ejemplo, puede configurarse para notificar a un administrador los intentos para tener acceso de entrada y para indicar a los usuarios por qué fueron negados sus intentos para tener acceso de salida.

Una desventaja de SOCKS es que funciona sólo para clientes basados en TCP; no funciona para clientes basados en UDP, quizá desee obtener otro programa: UDP Packet Relayer, que funciona en forma muy similar para los clientes basados en UDP a como lo hace SOCKs para los clientes basados en TCP. Al igual que SOCKS, UDP Packet Relayer está disponibles en Internet de forma gratuita.

La ventaja principal de SOCKS es su popularidad. Debido a que se emplea ampliamente, la utlización del servidor y clientes tipo SOCKS (por ejemplo, versiones de programas como FTP y Telnet que ya han sido convertidos a SOCKS de usuario) están disponibles de forma común, y es fácil encontrar ayuda. Esto puede ser un arma de dos filos; se han reportado casos en donde los intrusos a sitios con Firewalls han instalado sus propios clientes con conocimientos de SOCKS.

SOCKS contiene los siguientes componentes:

- El servidor SOCKS, que debe ejecutarse en un sistema UNIX, aunque se ha migrado a muchas variantes de UNIX
- La biblioteca para clientes SOCKS para máquinas UNIX
- Las versiones tipo SOCKS de varios programas estándar para clientes UNIX, como FTP y Telnet.

Qué pasa si no puede utilizar un proxy

Quizá encuentre que no puede emplear un proxy para un servicio por una de tres razones:

- No dispone de un servidor proxy
- Un proxy no asegura lo suficiente el servicio.
- No puede modificar el cliente y el protocolo no permite que utilice procedimientos modificados.

No hay servidor proxy disponible

Si el servicio puede operar con proxy pero no puede encontrar un servidor de procedimiento modificado o clientes modificados para su plataforma, siempre puede hacer usted mismo el trabajo. Modificar un programa cliente TCP normal para utilizar SOCKS es muy fácil. Mientras haya bibliotecas SOCKS disponibles para la plataforma que le interesa, es cosa de cambiar algunas llamadas de biblioteca y recompilar, pero de tener el código fuente para el cliente.

Un proxy no asegurará el servicio

Si necesita utilizar un servicio inherentemente inseguro, un proxy no puede hacer mucho por usted. Debe instalar una máquina víctima, y dejar que la gente ejecute el servicio ahí. Esto puede ser difícil si utiliza un host con doble acceso sin enrutamiento para hacer un Firewall donde todas las conexiones deben utilizar un proxy; la máquina víctima necesitará estar en el lado de Internet del host con doble acceso.

El uso de un servidor inteligente a nivel de aplicación, que filtre los comandos inseguros, podría ayudar; pero requiere extrema preocupación en implementar el servidor y puede hacer no funcionales las partes importantes del servicio.

No puede modificar cliente o procedimientos.

Hay algunos servicios que simplemente no tienen espacio para codificar procedimientos de usuario (por ejemplo, ping y traceroute). Por fortuna, los servicios que no permiten al usuario pasar ningún dato al servidor tienden a ser pequeños, tontos y seguros. Quizá pueda proporcionarlos de forma segura en el host bastión, dejando que los usuarios inicien sesiones con él pero dándoles un shell (intérprete de comandos) que sólo les permita ejecutar los servicios que operan con un proxy y que quiere ofrecer.

3.4 POLÍTICAS DE SEGURIDAD

La palabra política hace que mucha gente se acobarde, ya que evoca documentos impenetrables creados por comités desconocidos, los cuales de inmediato son ignorados por todos los participantes (excepto cuando son una buena excusa o arma).

La política de la que se a hablar en este tema es como loa política exterior de una nación. Puede analizarse en documentos, pero su propósito básico es establecer una dirección, una teoría de lo que intenta lograr.

No hay duda de que será un proceso lago y difícil establecer una política de seguridad, y es exactamente el tipo de tarea opuesta a la que casi todos los técnicos disfrutan. Elaborar una

política tampoco tiene que ser fastidioso como cree. La mayoría de los problemas de las políticas de seguridad las ocasiona la gente que intenta escribir una política de seguridad que parezca una política de seguridad, esto es, que esta en términos legales y palabras técnicas y diga cosas amenazantes sobre cómo les conviene comportarse a los usuarios.

Otro problema que padece la gente al intentar escribir políticas de seguridad es que tiene una idea firme sobre cómo debe ser la política, y se siente incómoda si la que establecen no sigue ese estándar. Hay mucha palabrería en el hecho de decir que la seguridad debe ser absoluta: debe tener un sólo sitio en el que nadie pueda penetrar sin autorización; donde todas las contraseñas son excelentes y nadie usa la contraseña de otro para nada.

Por otro lado si tiene muy poca seguridad, puede perder la organización en manos de abogados o atacantes, y lo que importa ahí es lo que hace, no lo que escribe. Escribir políticas maravillosas que no se pongan en práctica no lo salvará de la gente que intente penetrar su computadora, y por lo general tampoco de pleitos legales.

3.4.1 QUE DEBE TENER UNA POLÍTICA DE SEGURIDAD

Primero y más importante, la política de seguridad es una forma de comunicarse con los usuarios y los gerentes. Debe decirles lo que deben saber para tomar las decisiones que deben tomar al respecto a la seguridad.

Explicaciones

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

Lenguaje común

La mayoría de las personas no son abogados ni expertos en seguridad. Se sienten a gusto con descripciones informales. Puede tener miedo de escribir una política semejante porque podría parecer incómodamente informal y demasiado personal. Pero es más importante hacer su política amistosa y comprensible que hacerla precisa y de apariencia oficial. Escríbala como si la estuviera explicando a un amigo razonablemente brillante pero no técnico.

Autoridad para la aplicación

Escribir la política no es lo importante; la cuestión es vivir bajo ella. Esto quiere decir que cuando no se sigue la política, debe hacer lago para solucionarlo. Alguien debe ser responsible de hacer que esas soluciones ocurran, y la política debe especificar quién será esa persona y el rango general de correctivos. He aquí algunos ejemplos de lo que debe especificar una política de seguridad.

- Los administradores de ciertos servicios tienen la autoridad de revocar el acceso.
- Se pedirá a los administradores que se ocupen de cierto tipo de transgresiones.

- Las instalaciones que no llenen ciertos requisitos pueden excluirse de la red corporativa y del acceso por parte de gente externa a la red corporativa.

La política debe especificar quién va a decidir y dar indicaciones sobre la clase de sanciones que se pueden imponer. No debe especificar con exactitud qué pasará cuando algo suceda.

Análisis de temas de seguridad específicos

Dadas las diferencias entre las organizaciones, es difícil ser específicos sobre los temas importantes en materia de seguridad:

- ¿ A quién se le permite tener una cuenta en su sitio? ¿Tiene cuentas huéspedes?
- ¿Pueden compartirse las cuentas entre varias personas? ¿Qué pasa si una secretaria usa la cuenta de un ejecutivo para procesar correo electrónico de esa persona?
- ¿Cuándo pierde la gente el derecho de tener una cuenta y qué hacer al respecto?
- ¿Quién puede instalar módems para entrar a la red? ¿Es correcto que otras personas instalen módems para hacer llamadas externas?
- ¿Qué debe hacer la gente antes de conectar una computadora a la red principal?
- ¿Cómo se protegerá la información financiera?
- ¿Cómo se protegerá la información confidencial sobre la gente?
- ¿Qué tienen que hacer los usuarios para protegerse a sí mismos y al sitio?
- ¿Qué puede hacer la gente en Internet? ¿Pueden transferir archivos ejecutables al azar y ejecutarlos?
- ¿Quién puede conectar su sitio con redes externas y qué es una red externa?
- ¿Cómo va a tener acceso la gente que viaja?
- ¿Qué información de la compañía considera confidencial? ¿Cómo será protegida?

3.4.2 CÓMO CONFORMAR UNA POLÍTICA DE SEGURIDAD

¿Qué es su política de seguridad?

El primer paso para conformar una política de seguridad funcional para su sitio es decidir cuál es su opinión personal. Si ha administrado un sitio o tomado decisiones sobre seguridad, ha fortalecido una teoría interna sobre la seguridad, aunque nunca la haya articulado. Debe tener una compresión clara y explícita de lo que la política interna antes de poder discutir con personas asuntos relacionados a fin de escribir una política para su sitio.

¿Cuál es la política de seguridad de su sitio?

El segundo paso en aras de conformar una política de seguridad para su sitio es determinar qué es la política de seguridad para los demás. ¿Qué esperan los usuarios y los administradores que la seguridad haga por ellos? ¿Qué piensan de la forma en que se maneja la seguridad en la actualidad?

Cada sitio tiene al menos una política de seguridad. El problema es que la mayoría de los sitios tienen más que una; tal vez tantas como gente hay relacionada con las computadoras del sitio.

Cuando hable con la gente, aclárele la razón de sus preguntas. Preguntar sobre políticas de seguridad tiende a darle la impresión a las personas de que intenta hacerles un examen. Si obtiene una de esas reacciones, deje de hacer preguntas sobre políticas de seguridad y vuelva a explicarles lo que hace y por qué.

3.4.3 FACTORES EXTERNOS QUE INFLUYEN EN LAS POLÍTICAS DE SEGURIDAD

Su sitio no es del todo independiente. Hay elementos fuera de un centro de cómputo que influye en la política de seguridad. Incluyen requisitos legales, obligaciones contractuales y políticas organizacionales existentes.

Por ejemplo, su organización puede tener, también, obligaciones contractuales para proteger la información. Si tiene en su sistema información de consumidores o clientes, quizá sus contratos lo hagan protegerla.

Si su organización tiene un departamento jurídico, consúltelo. Si su organización no dispone de un área jurídica, consulte al gerente general. En cualquier caso, encuentre cualquier política ya escrita y revísela para ver si dice algo relevante para su seguridad.

Haga que se cumplan las decisiones sobre estrategias y políticas.

Las decisiones estratégicas deben ser entendidas y hechas por gerencias de alto nivel o nunca se implementarán con éxito. Si no tiene el apoyo de las gerencias de alto nivel para su seguridad, no tendrá seguridad, así de simple. ¿Por qué no tiene el apoyo de los gerentes de alto nivel?, Tal vez, porque desde su punto de vista, no ha solucionado sus preocupaciones.

Involucre a los afectados

Puede ser la persona que mejor comprenda las cuestiones técnicas, pero no necesariamente la persona que mejor comprenda las necesidades de su institución en conjunto. Las decisiones de estrategia y políticas deben tomarlas personas que trabajen en conjunto. No puede simplemente llegar con una política que le guste, mostrársela a un gran número de personas y hacer que la acepten. Aun si logra (lo cual puede ser más difícil que lograr que ayuden a tomar decisiones inteligentes) no la seguirán.

CAPÍTULO 4

IMPLEMENTACIÓN DE UN FIREWALL EN FES CUAUTITLÁN

4.1 FIREWALL PARA LA FES CUAUTITLÁN UTILIZANDO LA ARQUITECTURA DE SUBRED DE PROTECCIÓN.

En este capítulo nos ocuparemos de una configuración básica de un Firewall para FES Cuautitlán, esta configuración nos dará una idea de los recursos y actividades que necesitamos obtener y desarrollar respectivamente para la construcción de un Firewall y cómo armar las diferentes piezas con las que contamos.

Los servicios que se van a proporcionar por medio de este Firewall son algunos de los servicios que se mencionaron en el capítulo 1 de este trabajo y que son los más básicos: acceso de terminal, transferencia de archivos, correo electrónico, WWW y DNS.

Se utilizará una arquitectura de subred de protección la cual se tocó ampliamente en el capítulo 2 de este trabajo, la cual tal vez, es la más común en la actualidad. Esta arquitectura proporciona buena seguridad.

En esta arquitectura existen dos variantes, una con dos routers y otra con un solo router. La arquitectura de subred de protección de un solo router funciona casi tan bien como la de dos routers y es un poco más barata. Sin embargo necesitamos un router que pueda manejar el filtrado de paquetes tanto de entrada como de salida en cada interface, sin embargo, usaremos la arquitectura de dos routers.

Los componentes de este tipo de Firewall incluyen:

- Red de perímetro.
- Router externo.
- Router interno.
- Host bastión.

Además de las máquinas del Firewall, en nuestra configuración existen otras muchas máquinas a lo largo de los campus de las FES Cuautitlán tales como servidores internos que prestan servicios tales como:

- Servidores de correo.
- Servidores de Noticias Usenet.
- Servidor DNS
- Cliente de varios servicios de Internet.

Cada uno de estos servicios se proporciona directa (filtrado de paquetes) o indirectamente (servidor proxy ejecutados en el host bastión).

Vamos a suponer (para propósito de este ejemplo) que confiamos en los usuarios internos no tratan de burlar el Firewall y no hay ninguna necesidad particular de monitorear o tener acceso a sus actividades de Internet.

También tenemos que contemplar el hecho de que debemos usar direcciones IP correctamente asignadas y enrutadas para sus redes internas y de perímetro, de otra forma tendríamos que usar servidores proxy, ya que no podemos permitir la salida hacia Internet de paquetes con tales direcciones IP no asignadas correctamente.

Por último tenemos que utilizar números distintos de red para la red de perímetro y la red interna, a fin de detectar con facilidad los paquetes falsificados.

4.2 CONFIGURACIÓN DE SERVICIOS.

¿Cómo podemos proporcionar los servicios básicos de Internet con esta arquitectura?

Telnet

Podemos proporcionar Telnet de salida a través del filtrado de paquetes o de proxies. ¿Qué opción debemos de utilizar?

Proxy necesita clientes modificados o procedimientos de usuario modificados; ambos resultarán tediosos a quien los implemente. Proxy nos permite restringir o monitorear cómo usan Telnet los usuarios al forzarlos a autenticarse en un servidor proxy antes de completar sus solicitudes.

Es mucho más difícil proporcionar, con seguridad y conveniencia, Telnet de entrada. Si fuera necesario, podría proporcionarse en el host bastión empleando autenticación adicional.

FTP

A diferencia de Telnet, FTP no se presta a una solución simple utilizando filtrado de paquetes. Como el modo FTP normal necesita una conexión de entrada a un puerto arbitrario arriba del 1023, intentar permitirlo sin hacer nada más da a los atacantes acceso a toda clase de servicios ejecutados en nuestra red interna.

Así que tenemos dos opciones:

- Soportar el modo pasivo por medio del filtrado de paquetes.
- Soportar el modo normal por medio de proxy.

Un modo realista sería usar filtrado de paquetes y proxies, empleando una compuerta de acceso como ftp-gw, de TIS FWTK, que no requiere modificación en los clientes. Los clientes que soportan el modo pasivo trabajarán a través de los filtros de paquetes. En plataformas donde podemos remplazar con facilidad los clientes que utilizan nuestros usuarios, podemos proporcionar clientes en modo pasivo.

Por otra parte si quisiéramos monitorear el uso de FTP, o si estuviéramos usando un número de red no asignado o no enrutado, tendríamos que emplear proxy forzosamente, pero ese no es el caso. También usaríamos proxy si decidiéramos ocultar información DNS, lo que podría evitar el problema de falsificar información para revisiones posteriores.

Debemos tener en cuenta que si queremos usar el servidor proxy de TIS FWTK, ftp-gw, en un host bastión, nuestro sistema de filtrado de paquetes deberá permitir conexiones TCP de puertos arriba de 1023 en el host bastión hacia puertos arriba del 1023 en los servidores internos y del puerto 20 de servidores externos hacia puertos arriba del 1023 en nuestro host bastión, para canales de información FTP. Esto significa que alguien que entra al host bastión podría conectarse fácilmente con cualquier servidor en cualquier servidor externo que use un puerto TCP arriba del 1023.

Tenemos que recordar también, que bloquear puertos específicos en lugar de bloquearlos todos y después permitir puertos específicos, por lo general es una estrategia peligrosa. Ya que es difícil desarrollar y mantener una lista completa de puertos que deben bloquearse en un sitio dado.

El FTP anónimo de entrada es otra cuestión, y lo proporcionaremos. Cómo ya estamos usando TIS FWTK, podemos buscar seguridad en vez de características y usar el servidor FTP anónimo de TIS FWTK. Si vamos a proporcionar FT anónimo de manera importante, quizá deberíamos usar el servidor FTP wuarchie que tiene más elementos y ejecutarlo en una máquina que no sea el host bastión principal pero si el de la red de perímetro.

SMTP

No tenemos muchas opciones para SMTP en ninguna configuración. Queremos que todas las conexiones SMTP vayan a una sola máquina con un servidor SMTP seguro, y no confiamos en que cualquier máquina interna tenga servidores SMTP seguros. Esto quiere decir que pondremos un servidor SMTP seguro en el host bastión y usaremos registros MX de DNS para dirigir todo el correo de entrada en el host bastión, el cual después lo pasará a un solo servidor SMTP interno seguro.

Configuraremos SMTP, con el host bastión actuando como intermediario para el correo de entrada y salida:

- Publicar registros MX de DNS de forma tal que dirijan el correo que entra al sitio del host bastión.
- Configurar las máquinas internas de modo que manden todo el correo de salida al host bastión.
- Configurar el host bastión para que envíe todo el correo de entrada a un solo servidor de correo interno y para que envíe el correo de salida directamente a las máquinas destino.

HTTP

Al igual que para los otros servicios, para HTTP podemos usar el filtrado de paquetes o servidores proxy a fin de dar servicio a los clientes internos. El filtrado de paquetes permitirá que

nuestros usuarios tengan acceso a los servidores HTTP sólo en los puertos estándar; lo que les permitirá llegar a todos los servidores HTTP.

Por otro lado, ya hemos decidido permitir que los servidores internos creen conexiones de salida a cualquier puerto en o arriba del 1024 para permitir FTP de modo pasivo directamente de los servidores internos a los servidores externos. Eso permitirá el acceso a casi cualquier servidor HTTP. Utilizar únicamente filtrado de paquetes sólo nos permitirá puertos HTTP en puertos no estándar abajo del 1024, y de todos modos que esos puertos ya están reservados.

Si usamos el servidor HTTP del CERN como servidor proxy, también puede hospedar páginas web. Hacerlo así puede mejorar en gran medida el funcionamiento de:

Clientes HTTP

Obtienen páginas del caché sobre la red interna, en lugar hacerlo del servidor original sobre nuestra conexión de Internet.

Clientes diferentes de HTTP

Los clientes HTTP no ocuparán demasiado ancho de banda de nuestra conexión de Internet.

Tanto proxy (CERN) como el filtrado de paquetes pueden ser elecciones atractivas. En definitiva sería preferible proxy si no proporcionamos FTP de modo pasivo directamente. Por otro lado sería preferible el filtrado de paquetes si quisiéramos usar clientes que no tengan soporte para proxy, o si no quisiéramos ofrecer un servidor HTTP y no tuviéramos otros servicios utilizando proxy.

DNS

Podemos ofrecer un mejor servicio DNS a través de un Firewall si en la red se utilizan un par de servidores uno en el host bastión, otro en un servidor interno.

Así que el servidor DNS en el host bastión es un servidor secundario de nuestro dominio, y que el servidor primario está en un servidor interno.

4.3 REGLAS SOBRE EL FILTRADO DE PAQUETES

Hasta este momento de la configuración del Firewall hemos definido, los servicios de Internet que se van a prestar y el software necesario para funcionar. Por consiguiente, tenemos que definir las reglas de filtrado de paquetes que es necesario para soportar nuestra configuración.

Para hacer esto el sistema de filtrado de paquetes debe de:

- Que pueda distinguir entre paquetes de entrada y salida.
- Puede filtrar por dirección fuente, dirección destino, tipo de paquete (TCP o UDP), puerto fuente y puerto destino.
- Puede filtrar independientemente de si el bit ACK está encendido o no.

- Aplica en orden las reglas que se enlistaron.

Router interno

El propósito del router interno es proteger la red interna de Internet y de propio host bastión. El router interno necesita las siguientes reglas para soportar nuestra configuración.

REGLA	DIRECCIÓN	DIRECCIÓN FUENTE	DIRECCIÓN DESTINO	PROTO-COLO	PUERTO FUENTES	PUERTO DESTINO	ACK	ACCIÓN
Spoof	Entrada	Interna	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Prohibir
Telnet-1	Salida	Interna	Cualquiera	TCP	>1023	23	Cualquiera	Permitir
Telnet-2	Entrada	Cualquiera	Interna	UDP	23	1023	Sí	Permitir
FTP-1	Salida	Interna	Cualquiera	TCP	>1023	21	Cualquiera	Permitir
FTP-2	Entrada	Cualquiera	Interna	TCP	21	> 1023	Sí	Permitir
FTP-3	Salida	Interna	Cualquiera	TCP	> 1023	> 1023	Cualquiera	Permitir
FTP-4	Entrada	Cualquiera	Interna	TCP	> 1023	> 1023	Sí	Permitir
FTP-5	Salida	Interna	Bastión	TCP	> 1023	21	Cualquiera	Permitir
FTP-6	Entrada	Bastión	Interna	TCP	21	> 1023	Sí	Permitir
FTP-7	Entrada	Bastión	Interna	TCP	Cualquiera	6000-6003	Cualquiera	Prohibir
FTP-8	Entrada	Bastión	Interna	TCP	> 1023	> 1023	Cualquiera	Permitir
FTP-9	Salida	Interna	Bastión	TCP	> 1023	> 1023	Sí	Permitir
SMTP-1	Salida	Interna	Bastión	TCP	> 1023	25	Cualquiera	Permitir
SMTP-2	Entrada	Bastión	Interna	TCP	25	> 1023	Sí	Permitir
SMTP-3	Entrada	Bastión	Servidor SMTP Interno	TCP	> 1023	25	Cualquiera	Permitir
SMTP-4	Salida	Servidor SMTP Interno	Bastión	TCP	25	> 1023	Sí	Permitir
HTTP-1	Salida	Interna	Bastión	TCP	> 1023	80	Cualquiera	Permitir
HTTP-2	Entrada	Bastión	Interna	TCP	80	> 1023	Sí	Permitir
DNS-1	Salida	Servidor DNS Interno	Bastión	UDP	53	53	a	Permitir
DNS-2	Entrada	Bastión	Servidor DNS Interno	UDP	53	53	A	Permitir
DNS-3	Salida	Servidor DNS Interno	Bastión	TCP	> 1023	53	Cualquiera	Permitir
DNS-4	Entrada	Bastión	Servidor DNS Interno	TCP	53	> 1023	Sí	Permitir
DNS-5	Entrada	Bastión	Servidor DNS Interno	TCP	> 1023	53	Cualquiera	Permitir
DNS-6	Salida	Servidor DNS Interno	Bastión	TCP	53	> 1023	Sí	Permitir
Default-1	Fuera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Prohibir
Default-2	Entrada	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Prohibir

Veamos alguna información adicional sobre cada conjunto de reglas:

Spoof

Bloquea los paquetes de entrada que dicen tener direcciones IP internas (ve paquetes falsificados que cree son enviados por un atacante).

Telnet 1 y Telnet2

Permite conexiones Telnet de salida.

FTP-1 y FTP-2

Permiten conexiones de salida a servidores FTp, para que las usen los clientes internos en modo pasivo que interactúen directamente con tales servidores.

FTP-3 y FTP-4

Permiten conexiones del canal de datos FTP de clientes internos en modo pasivo a servidores FTP externos. Estas reglas en realidad permiten todas las conexiones de puertos TCP internos arriba del 1023 a puertos TCP externos arriba del 1023.

FTP-5 y FTP-6

Permiten que los clientes FTp internos (de modo no pasivo) abran un canal de comandos FTP al servidor proxy FTP en el host bastión. Estas reglas son redundantes si ponemos las reglas FTP-1 y FTP-2 antes, ya que el host bastión como fuente o destino (cubierto por FTP-5 y FTP-6) es un subgrupo de "Todo" (cubierto por las reglas FTP-1 y FTP-2)

FTP-7 y FTP-9

Permite conexiones de datos del servidor proxy en el host bastión a clientes internos no pasivos. La regla FTP-7 evita que un atacante que ha penetrado en el host bastión ataque a los servidores internos por medio del espacio creado por las reglas FTP-8 y FTP-9.

SMTP-1 y SMTP-2

Permiten correo de salida de servidores internos al host bastión.

SMTP-3 y SMTP-4

Permiten correo de entrada del host bastión a el servidor de correo interno.

HTTP-1 y HTTP-2

Permiten que clientes HTTP internos se conecten al servidor proxy HTTP en el host bastión.

DNS-1

Permite consultas y respuestas DNS basadas en UDP, del servidor DNS interno al servidor DNS del host bastión.

DNS-2

Permite consultas y respuestas de DNS basadas en UDP, del servidor DNS del host bastión al servidor DNS interno.

DNS-3 y DNS-4

Permiten consultas DNS basadas en TCP, del servidor DNS del host bastión al servidor DNS interno, así como respuestas a estas consultas.

DNS-5 y DNS-6

Permiten preguntas DNS basadas en TCP, del servidor DNS interno a los servidores DNS del host bastión, así como respuestas a esas preguntas. También permite transferencias de zona en las que el servidor DNS del host bastión es el servidor primario y servidor DNS interno es el secundario.

Default-1 y Default-2

Bloquean todos los paquetes que no estén permitidos específicamente por alguna de las reglas precedentes.

Router externo

El router externo tiene un doble propósito:

- Conectar la red de perímetro (y el sitio entero) al resto del mundo.
- Proteger la red de perímetro y la red interna del mundo exterior.

Si se puede establecer filtrado en el router externo, es buena idea hacerlo. Al menos puede servir como respaldo a una parte del filtrado en ese router. Para nuestra configuración necesitamos establecer las siguientes reglas:

REGLA	DIRECCIÓN	DIRECCIÓN FUENTE	DIRECCIÓN DESTINO	PROTO-COLO	PUERTO FUENTES	PUERTO DESTINO	ACK	ACCIÓN
Spoof-1	Entrada	Interna	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Prohibir
Spoof-2	Entrada	Perímetro	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Prohibir
Telnet-1	Salida	Interna	Cualquiera	TCP	>1023	23	Cualquiera	Permitir
Telnet-2	Entrada	Cualquiera	Interna	TCP	23	> 1023	Sí	Permitir
FTP-1	Salida	Interna	Cualquiera	TCP	>1023	21	Cualquiera	Permitir
FTP-2	Entrada	Cualquiera	Interna	TCP	21	> 1023	Sí	Permitir
FTP-3	Salida	Interna	Cualquiera	TCP	> 1023	> 1023	Cualquiera	Permitir
FTP-4	Entrada	Cualquiera	Interna	TCP	> 1023	> 1023	Sí	Permitir
FTP-5	Salida	Bastión	Cualquiera	TCP	> 1023	21	Cualquiera	Permitir
FTP-6	Entrada	Cualquiera	Bastión	TCP	21	> 1023	Sí	Permitir
FTP-7	Entrada	Cualquiera	Bastión	TCP	20	6000-6003	Cualquiera	Prohibir
FTP-8	Entrada	Cualquiera	Bastión	TCP	20	> 1023	Cualquiera	Permitir
FTP-9	Salida	Bastión	Cualquiera	TCP	> 1023	20	Sí	Permitir
FTP-10	Entrada	Cualquiera	Bastión	TCP	> 1023	20	Sí	Permitir
FTP-11	Salida	Bastión	Cualquiera	TCP	21	> 1023	Sí	Permitir
FTP-12	Salida	Bastión	Cualquiera	TCP	20	> 1023	Cualquiera	Permitir
FTP-13	Entrada	Cualquiera	Bastión	TCP	> 1023	20	Sí	Permitir
FTP-14	Entrada	Cualquiera	Bastión	TCP	> 1023	> 1023	Cualquiera	Permitir
FTP-15	Salida	Bastión	Cualquiera	TCP	> 1023	> 1023	Cualquiera	Permitir
SMTP-1	Salida	Bastión	Cualquiera	TCP	> 1023	25	Cualquiera	Permitir
SMTP-2	Entrada	Cualquiera	Bastión	TCP	25	> 1023	Sí	Permitir
SMTP-3	Entrada	Cualquiera	Bastión	TCP	> 1023	25	Cualquiera	Permitir
SMTP-4	Salida	Bastión	Cualquiera	TCP	25	> 1023	Sí	Permitir
HTTP-1	Salida	Bastión	Cualquiera	TCP	> 1023	Cualquiera	Cualquiera	Permitir
HTTP-2	Entrada	Cualquiera	Bastión	TCP	Cualquiera	> 1023	Sí	Permitir
DNS-1	Salida	Bastión	Cualquiera	UDP	53	53	a	Permitir
DNS-2	Entrada	Cualquiera	Bastión	UDP	53	53	A	Permitir
DNS-3	Entrada	Cualquiera	Bastión	UDP	Cualquiera	53	A	Permitir
DNS-4	Salida	Bastión	Cualquiera	UDP	53	Cualquiera	A	Permitir
DNS-5	Salida	Bastión	Cualquiera	TCP	> 1023	53	Cualquiera	Permitir

DNS-6	Entrada	Cualquiera	Bastión	TCP	53	> 1023	Si	Permitir
DNS-7	Entrada	Cualquiera	Bastión	TCP	> 1023	53	Cualquiera	Permitir
DNS-8	Salida	Bastión	Cualquiera	TCP	53	> 1023	Si	Permitir
Default-1	Salida	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Prohibir
Default-2	Entrada	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Prohibir

Spoof-1 y Spoof-2

Bloquean paquetes de entrada que dicen tener direcciones IP internas o de la red de perímetro (paquetes presuntamente falsificados).La regla spoof-2 es exclusiva del router externo.

Telnet-1 y Telnet-2

Permiten conexiones Telnet de salida.

FTP-1 a FTP-4

Permiten conexiones de salida FTP en modo pasivo y son idénticas a las reglas correspondientes en el router externo.

FTP5 y FTP-6

Permiten que en el servidor proxy FTP del host bastión abra un canal de comandos FTP a servidores de Internet.

FTP-7 a FTP-9

Permiten conexiones de información FTP de servidores FTP externos al servidor proxy del host bastión. La regla FTp-7 evita que un atacante ataque los servidores en el host bastión a través del hueco que crean las reglas FTP-8 y FTP-9.

FTP-10 a FTP-15

Permiten FTP en modo pasivo y en modo normal de clientes internos al servidor FTP anónimo en el host bastión. No hay reglas equivalentes en el router interno, ya que no hay servidores FTP en la red interna a la que puedan acceder los clientes externos.

SMTP-1 y SMTP-2

Permiten correo de salida del host bastión al mundo exterior.

SMTP-3 y SMTP-4

Permiten correo de entrada del mundo exterior al host bastión.

HTTP-1 y HTTP-2

Permiten que el servidor proxy HTTP del host bastión se conecte a los servidores HTTP de cualquier máquina en Internet. En realidad, estas reglas permiten que cualquier programa cliente TCP en el host bastión usando el puerto arriba del 1023 para contactar cualquier programa servidor en cualquier servidor de Internet empleando cualquier puerto.

HTTP-3 y HTTP-4

-Permiten que los clientes externos se comuniquen con el servidor HTTP de host bastión. No hay reglas equivalentes en el router interno porque no hay servidores HTTP en la red interna a la que puedan acceder los clientes externos.

DNS-1

Permite consultas y respuestas DNS basadas en UDP, del servidor DNS del host bastión a los servidores DNS del mundo exterior.

DNS-2

Permite consultas y respuestas DNS basadas en UDP, de los servidores DNS de Internet al servidor DNS del host bastión.

DNS-3 y DNS-4

Permiten que clientes DNS externos basados en UDP pregunten al servidor DNS del host bastión y que éste le responda.

DNS-5 y DNS-6

Permiten consultas DNS basadas en TCP, del host bastión a los servidores DNS de Internet, así como respuestas a esas consultas.

DNS-7 y DNS-8

Permiten consultas DNS basadas en TCP, del mundo exterior al servidor DNS del host bastión, así como respuestas a esas consultas.

Default-1 y Default-2

Bloquean todos los paquetes que no estén permitidos específicamente por alguna de las reglas precedentes, al igual que las reglas correspondientes en el router interno.

4.4 OTRAS TAREAS DE CONFIGURACIÓN

Existen otras tareas que se deben ejecutar después de haber hecho las reglas para el filtrado e paquetes:

En todas las máquinas internas

Configurar el correo electrónico para que sea enviado a el host bastión. También debemos instalar clientes FTP en modo pasivo.

En el servidor de correo interno

Instalar los programas smap y smapd de TIS FWTK y una versión actualizada del programa de correo para que tenga un servidor SMTP confiable.

En el servidor DNS interno (primario)

Colocar un registro MX por cada registro A, dirigiendo el correo de entrda al host bastión; el servidor de correo interno puede necesitar más registros MX para dirigir el tráfico internamente.

En el host bastión

Tenemos que hacer toda la configuración estándar del host bastión (quitar los servidores que se usan, agregar clave de acceso). Habilitar TIS FWTK y configurar FTP por medio de proxy, smap y smapd, y su servicio FTP. También activar el servidor HTTP del CERN y configurarlo para que haga proxy.

De acuerdo con las estrategias de seguridad que analizamos en el capítulo uno, podríamos hacer un análisis de que tan buena es nuestra configuración.

Veamos a continuación algunos de los aspectos:

Menor privilegio

Como observamos anteriormente el principio de menor privilegio dice que un objeto (programa, persona, router) debe tener los mínimos privilegios necesarios para realizar su tarea asignada, no más. En nuestra configuración, por ejemplo, configurar SMTP para que el correo de salida pase por el host bastión (en lugar de pasar de manera directa a los servidores en Internet) es una aplicación de menor privilegio, porque le permite tener un mayor control de cómo se conectan los sistemas internos con los externos.

Punto de choque

Este principio está aplicado (cuello de botella) está aplicado con claridad en nuestra configuración, porque todo lo que existe entre los clientes internos e Internet pasa a través de la red de perímetro. Otros muchos pasan a través del host bastión, por medio de proxy.

Diversificación de la defensa

Cuando utilizamos routers de distintos proveedores para los sistemas de filtrados de paquetes interno y externo, estamos aplicando el principio de diversificación de la defensa.

CONCLUSIONES

Según algunas encuestas se estima que cerca de 30,000 redes de computadoras están conectadas ahora a Internet, lo cual se traduce en que casi 10,000,000 de computadoras y algo así como 30,000,000 usuarios a través del mundo, los cuales son "ciudadanos activos" de esta la llamada Supercarretera de la información.

Hay, sin embargo, algunos aspectos en los cuales se tiene que tener consideración antes de conectarse a Internet tales como: ¿Qué tipo de personas están conectadas a la red? ¿Es Internet confiable y seguro? ¿Qué servicios y recursos ofrece Internet? ¿Qué precauciones debo tomar?

En vista de lo anterior podemos decir que seguramente existen razones de peso para estar preocupados por las consecuencias que uno puede tener al conectarse a Internet sin estrategias confiables de seguridad que nos permitan conciliar el sueño, cuando no estemos navegando en la red.

Por supuesto, se puede decir que no existen los sistemas completamente seguros, sin embargo, el riesgo que se corre puede se reducido significativamente. De ahí que probablemente el nivel de seguridad va de acuerdo al valor de la información almacenada en los sistemas, la cual puede ser atacada por personas ajenas a nuestra organización.

En consecuencia si queremos conectarnos a Internet o ya lo estamos, tenemos que estudiar cuáles son las mejores alternativas en lo que a seguridad se refiere y poder aprovechar al máximo las mezclas de tecnologías. De ahí que los Firewalls se constituyan en nuestro tiempo como una de las formas más confiables de poner una "pared de fuego" a los "curiosos" que intentan internarse en nuestro sitio sin autorización alguna.

Sin duda, al estar implementado las diversas tecnologías disponibles, podremos estar tranquilos de que nuestra información no será una zona fácil de atacar, y como consecuencia podremos conciliar el sueño.

BIBLIOGRAFÍA

Wack p. John, Carnahan Lisa — Keeping Your Site Comfotably Secure
NIST Special Publications 800-10
U.S DEPARTMENT OF COMMERCE

Chapman Brent, Zwicky Elizabeth — BUILDING INTERNET FIREWALLS
Editorial O'REILLY. U.S.A. 1997.

URL´s

Galvin, Peter — Firewalls in many flavors
www.sunworld.com/sunworldon...ol-01-1996/swol-01-security_p.html

SunWorld Online — Watch your back door
/sunworld/online/swol-12-1995/swol-12-security.html

RESOURCES
ftp.greatcircle.com
news:comp.security.firewalls

TIS TOOLKIT
www.tis.com/HOME/NetworkSecurity.html

SOCKS Whitepaper
www.socks.nec.com./SOCKS5Info.html

SOCKS
ftp.nec.com/pub/security

Stahs of Firewalls Whitepapers
www.tis.com/HOME/NetworkSecurity/Firewalls/Firewalls.html