ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación
Licenciatura en Sistemas de Información

"ESTUDIO DE LAS NECESIDADES TECNOLÓGICAS PARA LA IMPLEMENTACIÓN DE SERVIDORES WEB SEGUROS"

TÓPICO DE GRADUACIÓN

Previa la obtención del Título de:

LICENCIADO EN SISTEMAS DE INFORMACIÓN

Presentado por

Lourdes Isabel Alvarado Chamaidán
Simón Bolívar Bravo Sánchez
Jessica Cecilia González Cevallos

Guayaquil – Ecuador

2005

AGRADECIMIENTO

Al DIOS todopoderoso quien siempre está en mi vida ayudándome en todo sentido, material y espiritual, en especial en la culminación de mi carrera.

Lourdes Isabel Alvarado Chamaidán

AGRADECIMIENTO

Agradezco a DIOS por haberme dado la vida y por la oportunidad de tener una familia comprensiva y unida, que siempre me han brindado el apoyo necesario en todas las etapas de mi vida y me han motivado a seguir adelante en cada tropiezo que se me ha presentado a lo largo de mi carrera.

Simón Bolívar Bravo Sánchez

AGRADECIMIENTO

A la persona en la que siempre puedo confiar y no me abandona nunca me da pruebas para hacerme mas fuerte y poder vencer cualquier obstáculo que se presente, esa persona es DIOS, a mis padres, mis hermanas y mi sobrino por haberme brindado cariño y motivación para alcanzar mis sueños.

Jessica Cecilia González Cevallos

DEDICATORIA

A las personas mas especiales en mi vida Héctor y Fresia, mis padres. Iris y Ely mis hermanas. Claudio mi novio y mis lindos sobrinos Xavier y Sandro, ya que son la inspiración a todas mis metas y planes.

Lourdes Isabel Alvarado Chamaidán

DEDICATORIA

A mi madre por ser mi principal punto de apoyo y comprensión en este camino. A mis hermanos, los ausentes y presentes, por haber sembrado en mí la semilla de la superación; este triunfo es fruto de su invaluable ayuda. A mi esposa, Marina, por haberme dado su apoyo para alcanzar esta meta, por brindarme su guía y consejos en el momento oportuno. A mi hijo, para sembrar en él la fuerza de voluntad necesaria para lograr sus más grandes sueños.

Simón Bolívar Bravo Sánchez

DEDICATORIA

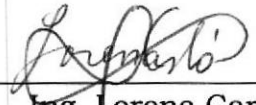
A las personas mas importantes en mi vida, mis padres Edith y Ober, por ser quienes en todo momento me han brindado su amor y confianza, por ser la motivación de seguir por el camino del éxito, para ustedes una muestra de mi esfuerzo y gratitud por su amor incondicional.

Jessica Cecilia González Cevallos

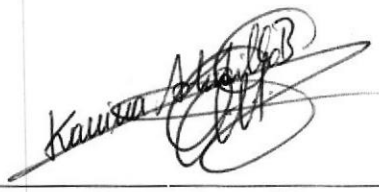
TRIBUNAL DE GRADO

Ing. Albert Espinal Santana
DIRECTOR DE TESIS

Ing. Mónica Villavicencio
PRESIDENTA DEL TRIBUNAL



Ing. Lorena Carlo
MIEMBRO PRINCIPAL



Ing. Karina Astudillo
MIEMBRO PRINCIPAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

Lourdes Isabel Alvarado Chamaidán

Simón Bolívar Bravo Sánchez

Jessica Cecilia González Cevallos

RESUMEN

En los últimos años todos los medios de difusión han hecho eco del futuro de las autopistas de la información, cuyo embrión está representado por la red Internet. Que con el gran crecimiento que ha tenido permite mayores formas de ataque a la seguridad en red, incluyendo los virus, Caballos de Troya y penetración de las redes internas.

Los servidores Web son designados para recibir solicitudes anónimas desde auténticos hosts en la Internet y a liberar las solicitudes de información en una manera rápida y eficiente. De tal forma, ellos proveen un portal que puede ser usado por amigos y enemigos igualmente. El tipo de tecnología que mejor cumple con estas demandas se deduce a través de estudios que se realizan para la implementación de servidores Web seguros. El presente trabajo pretende contribuir a este fin.

En este documento se especifica el análisis y los fundamentos teóricos y técnicos para elegir un Servidor Web, así como también las herramientas necesarias para que el funcionamiento del mismo sea el que se espera. También incluye procedimientos necesarios para la seguridad del Servidor Web como tal y como establecer políticas que nos ayuden en esta tarea.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Pág.

INTRODUCCIÓN

En los últimos años todos los medios de difusión han hecho eco del futuro de las autopistas de la información, cuyo embrión está representado por la red Internet. Que con el gran crecimiento que ha tenido permite mayores formas de ataque a la seguridad en red, incluyendo los virus, Caballos de Troya y penetración de las redes internas.

A raíz de la interconexión del mundo empresarial a esta red, viaja por ella y se almacena información de todo tipo, que abarca desde noticias, documentos, normas y aplicaciones informáticas de libre distribución hasta complejas transacciones que requieren medidas de seguridad que garanticen la confidencialidad, la integridad y el origen de los datos.

Los servidores Web son designados para recibir solicitudes anónimas desde auténticos hosts en la Internet y a liberar las solicitudes de información en una manera rápida y eficiente. De tal forma, ellos proveen un portal que puede ser usado por amigos y enemigos igualmente. El tipo de tecnología que mejor cumple con estas demandas se deduce a través de estudios que se realizan para la implementación de servidores Web seguros. El presente trabajo pretende contribuir a este fin.

CAPÍTULO 1

1. SERVIDORES WEB - EVALUACIÓN Y SELECCIÓN

1.1 Los Servidores Web

Los Servidores Web suministran páginas Web a los navegadores que lo solicitan como por ejemplo, Netscape Navigator, Internet Explorer de Microsoft. En términos más técnicos, los servidores Web soportan el Protocolo de Transferencia de Hipertexto conocido como HTTP (HyperText Transfer Protocol), el estándar de Internet para comunicaciones Web.

Usando HTTP, un servidor Web envía páginas Web en HTML y CGI, así como otros tipos de scripts a los navegadores o browsers cuando éstos lo requieren.

Cuando un usuario hace clic sobre un enlace (link) a una página Web, se envía una solicitud al servidor Web para localizar una pagina web determinada. El servidor Web recibe esta solicitud y suministra los datos que le han sido solicitados (una página HTML, un script interactivo, una página Web generada dinámicamente desde una base de datos) o bien devuelve un mensaje de error.

Los Servidores Web son aquellos que permiten a los clientes compartir datos, documentos y multimedia en formato Web. Aunque es parte de la

tecnología Cliente-Servidor, el servidor Web aporta algunas ventajas adicionales; como acceso más simple a la información (con un simple clic).

En el sentido más estricto, el término cliente/servidor describe un sistema en el que una máquina cliente solicita a una segunda máquina llamada servidor que ejecute una tarea específica.

El programa cliente cumple dos funciones distintas: por un lado gestiona la comunicación con el servidor, solicita un servicio y recibe los datos enviados por aquél. Por otro, maneja la interfaz con el usuario: presenta los datos en el formato adecuado y brinda las herramientas y comandos necesarios para que el usuario pueda utilizar las prestaciones del servidor de forma sencilla.

El programa servidor en cambio, básicamente sólo tiene que encargarse de transmitir la información de forma eficiente. No tiene que atender al usuario. De esta forma un mismo servidor puede atender a varios clientes al mismo tiempo.

La mayoría de servidores añaden algún nivel de seguridad a sus tareas. Por ejemplo, si usted ha ido a alguna página y el navegador presenta una ventana de diálogo que pregunta su nombre de usuario y contraseña, ha encontrado una página protegida por contraseñas. El servidor deja que el dueño o el administrador del servidor mantenga una lista de nombres y

contraseñas para las personas a las que se les permite ver la página, y el servidor deja que sólo esas personas quienes saben la contraseña tengan acceso.

Los servidores más avanzados añaden seguridad para permitir una conexión encriptada entre el servidor y el navegador, así la información más importante como números de tarjetas de crédito puede ser enviada por Internet.

Así mismo, los Servidores de páginas Web tienen un apartado muy importante de seguridad, permitiendo a sus administradores configurarlos de forma que restrinjan el acceso a programas malintencionados como virus y gusanos. Todos han de implementar:

- Políticas de acceso.
- Políticas de seguridad.
- Capacidad de envíos seguros. Apoyándose en la criptografía y el sistema de cifrado de clave pública.

El servidor Web es la parte fundamental en el desarrollo de las aplicaciones Web que vayamos a construir, ya que se ejecutarán en él.

Una de las misiones de los servidores Web es la de configurar una serie de directorios como los lugares donde se almacenan los scripts, de forma que las páginas que contengan enlaces puedan encontrarlos y ejecutarlos.

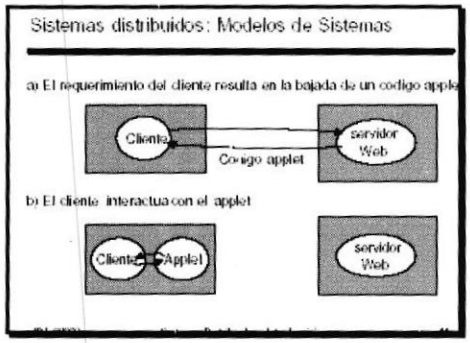


Figura 1.1 Modelo de los Sistemas Distribuidos

1.1.1 ¿Qué se debe analizar al elegir un Servidor Web?

Al momento de elegir un servidor es necesario analizar lo siguiente:

- **Estabilidad:** para evitar tiempo fuera de servicio.
- **Seguridad:** para evitar el hackeo del servidor.
- **Fácil auditoría:** permitirán llevar un control de los visitantes de nuestro sitio y sus preferencias, origen de las visitas, páginas más visitadas, etc.

1.1.2 Importancia de los Servidores Web

En la actualidad los servidores de información son los más usados en Internet. Esta popularidad ha obligado a muchas empresas y organismos a prestar sus servicios a través de estos medios.

Este es el motivo por el cual el desarrollo de nuevas aplicaciones basadas en Internet han experimentado un gran impulso. Uno de los

principales problemas a la hora de ofrecer servicios basados en el Web ha sido el gran esfuerzo necesario para mantener actualizada la información publicada en los servidores Web.

Un primer paso para reducir este esfuerzo es conseguir que las páginas HTML sean generadas total o parcialmente de forma dinámica, permitiendo el cambio de la información de forma automática e inmediata.

El hecho de generar dinámicamente los contenidos de un servidor de información presenta dos ventajas a la hora de administrarlo. En primer lugar se evita la necesidad de reeditar las páginas HTML estáticas cada vez que se produce un cambio en la información.

Y como segunda ventaja, se encuentra el hecho de la desaparición de las páginas con información obsoleta.

1.1.3 La expansión de los Sitios Web

En la actualidad hay más de 50 millones de sitios Web en el Internet[1] de acuerdo al estudio realizado en mayo del 2004 donde exactamente 50'550.965 sitios dieron respuestas. Un periodo de crecimiento reavivado para el Internet, viniendo sólo 13 meses detrás del estudio pasó la marca de 40-millones en abril, 2003.

---

[1] http://news.netcraft.com/archives/2004/05/03/may_2004_web_server_survey_finds_50_million_sites.html

Como comparación se tomaron 21 meses para ver que la Web se extendió de 30 millones a 40 de millones de sitios.

Mayo fue el décimo sexto mes consecutivo de crecimiento para la Web que tomó dos años para absorber el derrumbamiento del punto-com e industrias del telecomunicaciones.

La tendencia ascendente reasumió en febrero del 2003, cuando se descubrieron 35.8 millones de sitios (investigación realizada por NETCRAFT[1]); sobre el mismo número como en el estudio de diciembre del 2001.

El resultado del gran número de sitios es la recuperación de la economía a través de Internet, surgiendo así compañías viables y modelos de negocios.

Los recientes meses han visto informes de crecimiento fuerte por los anuncios online, sitios de suscripción pagados, los detalles de gastos online, e incluso las inversiones importantes y los contratos punto-com.

En las primeras investigaciones realizadas por Netcraft en agosto del 1995 se encuentran a 18,957 organizadores. Se alcanzaron resultados anteriores en el estudio en 1997 de abril (1 millones de sitios), el 2000 de

[1] www.netcraft.com

febrero (10 millones), el 2000 de septiembre (20 millones) y el 2001 de julio (30 millones).

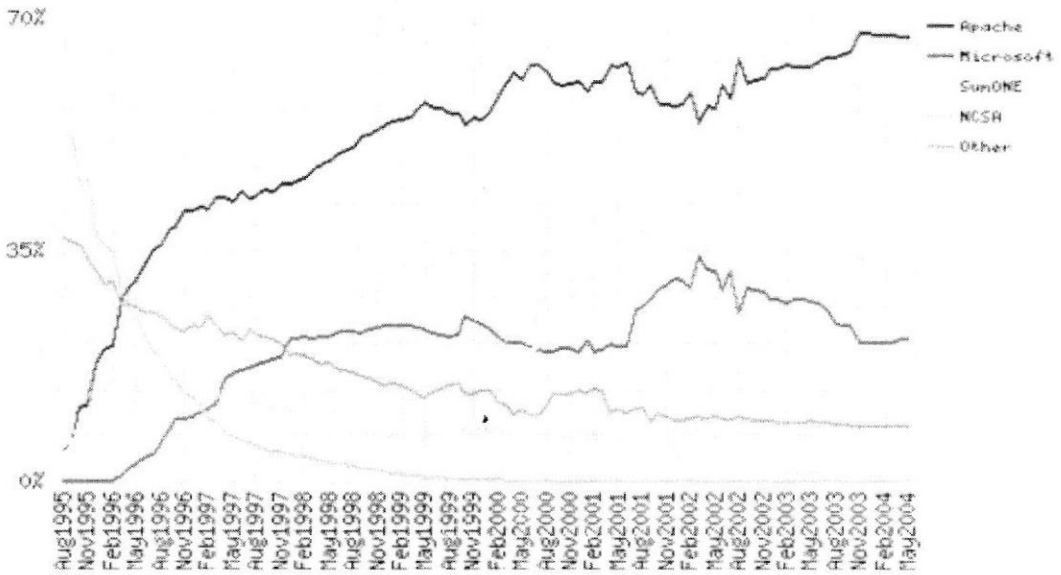


Figura 1.2 Porción del mercado para Servidores Agosto 1995 – Mayo 2004[1]

1.1.4 Posición de Apache en el estudio realizado por Netcraft

Apache ampliamente mantuvo su porción el mes de mayo con una ganancia de tres cuartos de millón de sitios.

La primera Conferencia Europea de apache fue anunciada recientemente, con Apachecon (nombre de la conferencia) que se sostuvo en Londres del 23-25 de octubre de 2004. Weblogic gana unos 90,000 sitios que dan alcance a thttpd y Zeus.

---

[1] http://news.netcraft.com/archives/2004/05/03/may_2004_web_server_survey_finds_50_million_sites.html

Microsoft anunció una nueva herramienta apuntada a dar la capacidad a Windows de producir rápidamente sitios Web. La herramienta ha sido codesarrollada con Interland.net que es actualmente uno de los hosts más grandes de Windows 2000 en EE.UU., con alrededor de 2,500 sitios. Interland también tiene alrededor de 30,000 hosts de los sitios en Linux, y unos 60,000 en NT4.

Durante los últimos dos años, gracias a los estudios realizados se ha podido comprobar un crecimiento significativo de los servidores web.

En estos días la práctica más común en las compañías que ofrecen servicios de Internet es incrementar servidores web, con el fin de poner un sitio como ejemplo o plantilla en la Web para cada dominio que ellos registran.

Tomando en cuenta que en los primeros días del Internet, los hosts eran los que mantenían los mejores indicadores de volumen de información evaluando la información y servicios que proporcionaban a la comunidad del mundo de Internet, en la actualidad la situación se mancha considerablemente, debido a que, ahora la Web incluye mucha actividad, así como también una cantidad inimaginable de sitios intactos en los cuales intervino la mano humana, esto produce automáticamente que cada vez aumente de manera visible la adquisición de hosts por parte de la comunidad.

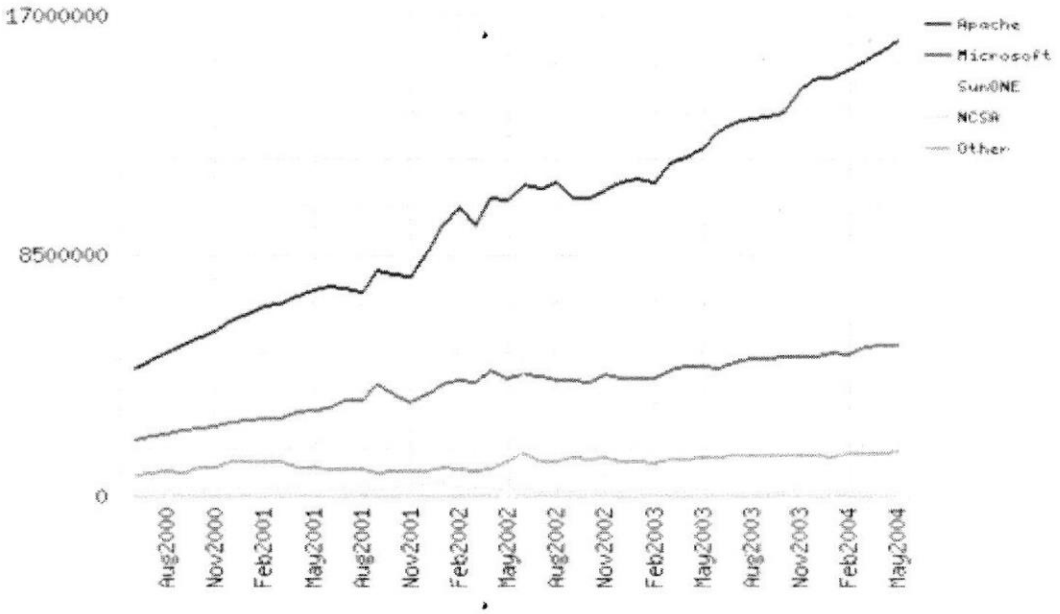


Figura 1.3 Servidores activos de todos los dominios Jun 2000 – May 2004[1]

Servidores	Abril 2004	%	Mayo 2004	%	Diferencia
Apache	15747757	69.01	16064821	69.26	0.25
Microsoft	5307932	23.26	5328368	22.97	-0.29
Sun	197204	0.86	189854	0.82	0.04
Zeus	262671	1.15	263124	1.13	-0.02

Tabla 1.1 Sitios Activos Abril 2004 – Mayo 2004

Entre los años 1996 y 1997, el número de direcciones de IP distintas habría sido una aproximación buena del número de sitios reales, desde que las compañías de hosting asignarían una dirección de IP típicamente a cada sitio con volumen distinto, y los nombres de dominio múltiples podrían apuntar a la dirección de IP para tener acceso a los sitios Web.

[1] http://news.netcraft.com/archives/2004/05/03/may_2004_web_server_survey_finds_50_million_sites.html

Sin embargo, con la adopción de HTTP/1.1 hosting virtuales, y la disponibilidad de carga que equilibra la tecnología es posible organizar fiablemente los sitios activos en uno solo, en un número pequeño de las direcciones IP. Por ejemplo FreeServe tiene alrededor de 150,000 hosts de los sitios en cuatro, cargas las direcciones de IP equilibradas.

En Noviembre del 2004 hubo respuesta de 56,115,015 sitios. El Internet ha crecido a 10.1 millones de sitios en los primeros 11 meses del año, incluyendo una ganancia de 726,549 sitios el último mes.

El 2004 debería terminar como el segundo año mas fuerte en el crecimiento del Internet, arrastrados desde el 2000, cuando la encuesta sumó 16.1 millones de sitios. La encuesta sumo 10.06 millones de sitios en el 2001 y 10.4 millones en el 2003.

Predominando la tendencia de Apache continúa en el mercado como el mayor Servidor Web, los porcentajes son considerablemente visibles.

Servidor	Octubre 2004	%	Noviembre 2004	%	Diferencia
Apache	37620349	67.92	38028642	67.77	-0.15
Microsoft	1685325	21.09	11923566	21.25	0.16
Sun	1685325	3.04	1761705	3.14	0.10
Zeus	748561	1.35	739006	1.32	-0.03

Tabla 1.2 Sitios Activos Octubre 2004 – Noviembre 2004

Este giro contribuye a la expansión de la creciente del e-commerce y banca en línea, tanto como precios bajos para los nombres de dominios.

El registrador de dominios y las compañías de hosting dicen que los pequeños negocios han sido compradores activos de dominios y sitios Web, aún cuando los beneficios del Internet como una herramienta de negocio ha sobrellevado temores acerca de la seguridad en el Internet, lo cual es a menudo citado como un factor en la adopción tardía de la Web por algunos pequeños negocios.

Durante el 2004 la Web ha emulado el modelo del mercado financiero de "Escalar una pared de preocupaciones" con un promedio de 911,000 sitios nuevos al mes, todo esto a pesar de la ola de incidentes de seguridad.

Figura 1.4 Participación de Hostnames y Sitios Web en el mercado Agosto 1995 – Diciembre 2004[1]

[1] http://news.netcraft.com/archives/2004/05/03/may_2004_web_server_survey_finds_50_million_sites.html

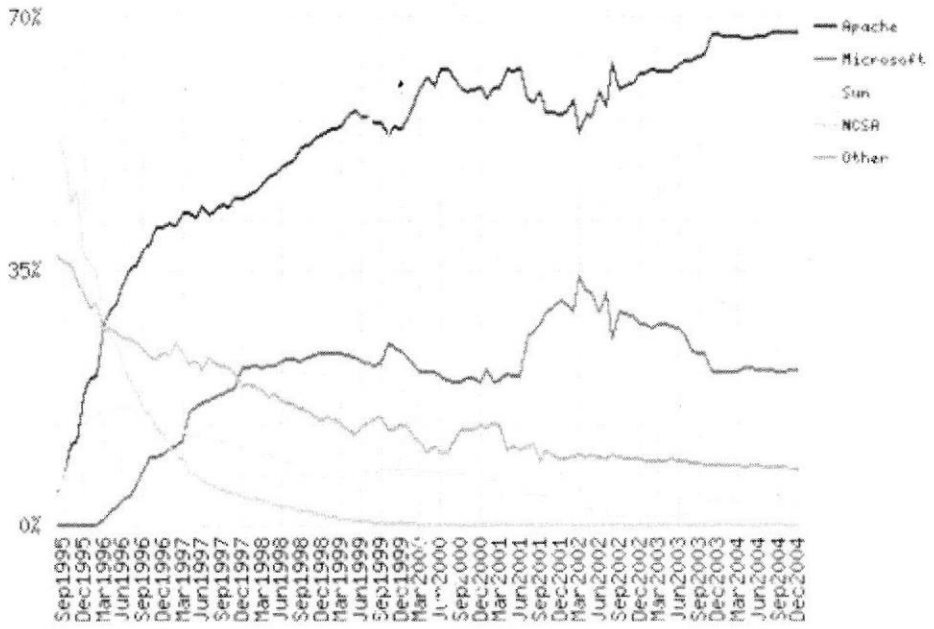


Figura 1.5 Participación de Servidores Web en el mercado Agosto 1995 – Diciembre 2004[1]

Servidor	Noviembre 2004	%	Diciembre 2004	%	Diferencia
Apache	38028642	67.77	38614673	67.84	0.07
Microsoft	11923566	21.25	12062761	21.19	-0.06
Sun	1761705	3.14	1812966	3.18	0.04
Zeus	739006	1.32	687508	1.21	-0.11

Tabla 1.3 Sitios Activos Noviembre 2004 –Diciembre 2004

En diciembre del 2004 se tuvo un total de 56,923,737 servidores.

[1] http://news.netcraft.com/archives/2004/05/03/may_2004_web_server_survey_finds_50_million_sites.html

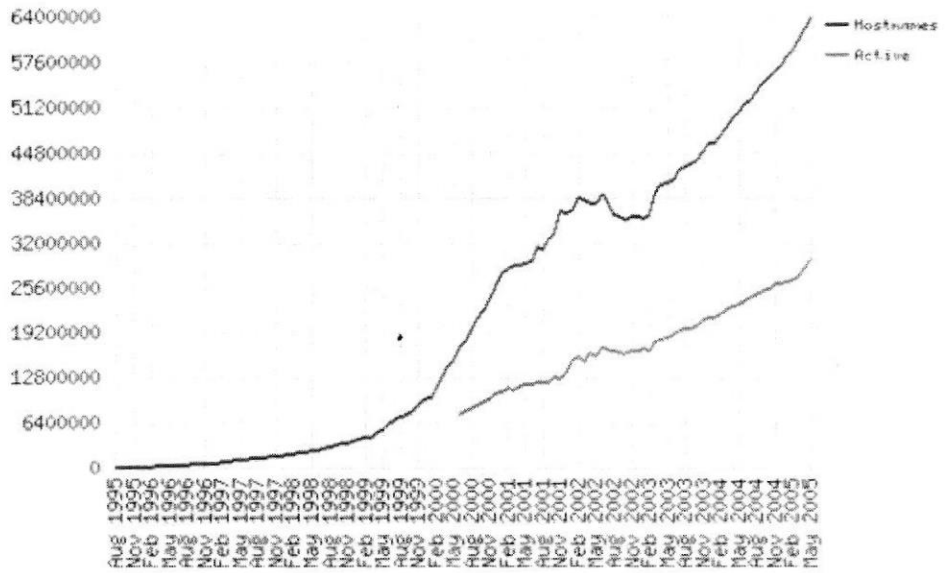


Figura 1.6 Participación de Hostnames y Sitios Web en el mercado Agosto 1995–Mayo 2005[1]

Servidor	Abril 2005	%	Mayo 2005	%	Diferencia
Apache	43174442	69.32	44072262	69.37	0.05
Microsoft	12735588	20.45	13049346	20.54	0.09
Sun	1880921	3.02	1856222	2.92	-0.10
Zeus	576582	0.93	562614	0.89	-0.04

Tabla 1.4 Sitios Activos Abril 2005 –Mayo 2005

1.1.5 Conclusión

En Mayo del 2005, como resultado de la encuesta se tuvo **63,532,742** sitios. El aumento fue de 1.24 millones de sitios en comparación con el mes de Abril. Continúa el potente crecimiento de Internet como medio para la comunicación y el comercio, lo cual demuestra un incremento promedio de 1.2 millones de sitios por mes en el transcurso del año 2005 a pesar de las constantes amenazas de seguridad.

[1] http://news.netcraft.com/archives/2004/05/03/may_2004_web_server_survey_finds_50_million_sites.html

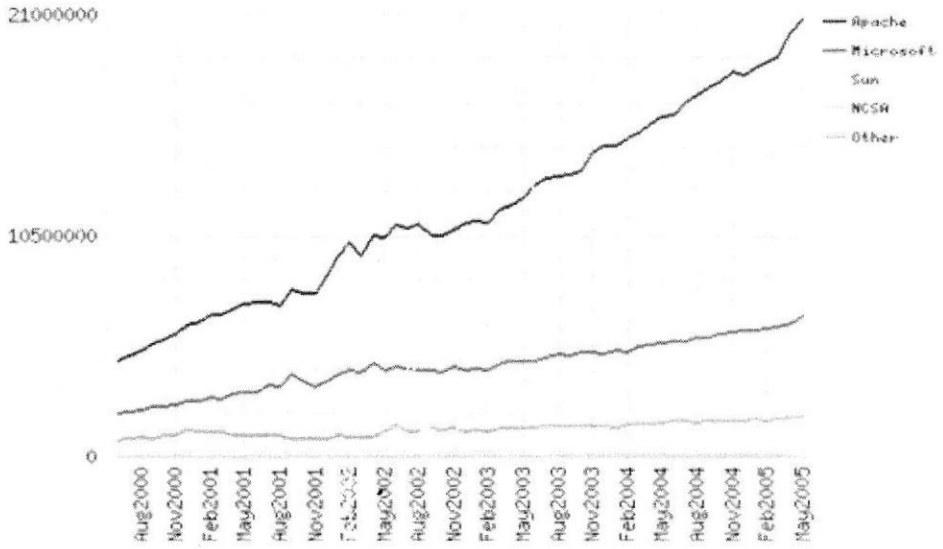


Figura 1.7 Participación de Servidores Web en el mercado Agosto 1995 – Mayo 2005[1]

1.1.6 Variaciones Regionales

Generalmente, la mayoría de los países pierden menos sitios como resultado de aplicar la metodología de los sitios activos que en EEUU. Adicionalmente, en muchos países el registro del nombre del dominio es más caro o más burocrático que en el EE.UU. ya que se ha restringido el uso de dominio múltiple a través de negocios. Alemania y el Reino Unido son los hosts más grandes de sitios externo al EE.UU., y las compañías del hosting acostumbran a usar tecnología ampliamente similar a la de EEUU.

[1] http://news.netcraft.com/archives/2005/05/01/may_2005_web_server_survey.html

CAPÍTULO 2

2. EVALUACIÓN Y SELECCIÓN DEL SISTEMA OPERATIVO

Hacer una verdadera comparación entre un sistema operativo y otro es un proceso muy delicado, en especial si se trata de ser lo más imparcial posible y a la vez tratar de abarcar los requerimientos que la mayoría de usuarios buscan de su plataforma, teniendo en cuenta esta reflexión lo primero que es necesario preguntarse antes de empezar es ¿cuáles son las bases para la comparación?, la respuesta a esta pregunta ha sido dividida en dos aspectos: administrativo y técnico. A nivel administrativo se detectó que la mayoría de usuarios sin conocimientos técnicos acerca del funcionamiento de los sistemas operativos evalúan 4 ítems:

- Esquema Licenciamiento y Precio
- Estabilidad y Desempeño
- Facilidad de uso
- Soporte

Por lo general estos son los principales aspectos que un administrador de sistemas de información tiene en cuenta al momento de presentar una evaluación a usuarios no técnicos (gerentes, jefes de división, etc.) cuando se trata de tomar una decisión en torno a cual plataforma adoptar para una empresa o proyecto.

A nivel técnico los tópicos que se han evaluado son:

- Compatibilidad con Otras Plataformas
- Portabilidad
- Requerimientos Hardware

Aunque a nivel técnico se encuentran gran cantidad de factores de más que no se incluyen en esta revisión es necesario aclarar que es tan sólo un caso particular y que procura ser muy genérico, cada compañía puede tener una perspectiva diferente en cuanto a la evaluación.

Ahora que se han explicado los componentes de la comparación, es indispensable enumerar a su vez los «sujetos de estudio», es decir los sistemas operativos sobre los que se pretende hacer este estudio, cabe destacar que al igual que los anteriores aspectos, estos también tienen la característica de la generalidad, por lo cual sólo se ha escogido una pequeña muestra de la gran cantidad disponible, estos son:

- FreeBSDTM
- GNU/Linux
- Mac OS X
- NetBSD
- OpenBSD
- Windows 2000
- Windows XP

A continuación se dará una breve descripción de cada uno de estos sistemas, con el objetivo de dar primero una visión general, después de esto se pasará a explicar cada uno de los tópicos de la comparación y dentro de estas se pasará de la generalidad a la especificidad.

2.1 Sistemas Operativos

2.1.1 FreeBSD

Figura 2.1 Logo FreeBSD[1]

FreeBSD es un sistema operativo derivado del 386 BSD, es un sistema operativo libre (y gratuito) creado por cientos de desarrolladores, es altamente usado como servidor de Internet debido a sus altas prestaciones en comunicaciones.

2.1.2 GNU/Linux

Figura 2.2 Logo GNU Linux[2]

[1] BSD Daemon Copyright 1988 by Marshall Kirk McKusick. All Rights Reserved

[2] The term "Linux" is a registered trademark of Linus Torvalds, the original author of the Linux kernel

Aunque se hablará más en profundidad acerca de este sistema operativo mas adelante, cabe destacar que es un sistema libre desarrollado por miles de personas a través de Internet, y que es el sistema operativo de mayor crecimiento en la actualidad, y el segundo de mayor uso en el mundo (tomando como uno sólo a toda la gama de la familia Windows.

2.1.3 Mac OS X

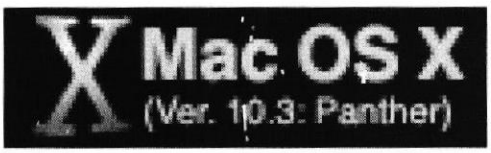


Figura 2.3 Logo Mac OS X[1]

Es un sistema operativo desarrollado por Apple Computer Inc., es una reescritura prácticamente completa de su sistema operativo Macintosh.

Este nuevo sistema Mac está basado en el sistema Darwin, un proyecto «OpenSource» (aunque esta es la definición que da Apple, de aquí en adelante utilizaremos el termino «Libre»), el cual utiliza características de otros sistemas UNIX como Mach, FreeBSD y otros, la idea detrás de Mac OS X.

El objetivo de Darwin es crear un completo sistema operativo con la flexibilidad y robustez de un UNIX y la facilidad de uso que siempre a caracterizado a los MAC.

[1] Mac OS X is a trademark of Apple Computer, Inc

2.1.4 NetBSD

Figura 2.4 Logo NetBSD[1]

NetBSD es tal vez el sistema operativo más portado en el mundo, es otro descendiente de 4.4 BSD y 386 BSD, por lo cual también se distribuye bajo los términos de la licencia BSD4.1 , lo que implica que puede ser libremente distribuido en forma binaria o de código fuente. El principal objetivo de NetBSD es la portabilidad, obviamente sin descuidar seguridad y estabilidad como la mayoría de derivados UNIX.

2.1.5 OpenBSD

Figura 2.5 Logo OpenBSD[2]

Este sistema operativo es derivado de NetBSD, sus principales metas son la seguridad, la estandarización y la portabilidad, está catalogado como

---

[1] "NetBSD" is a registered trademark of the NetBSD Foundation.

2 OpenBSD is a registered trademark of Theo de Raadt

el sistema operativo más seguro del mundo (aunque en términos de seguridad no se puede hablar de una verdad absoluta).

2.1.6 Windows 2003

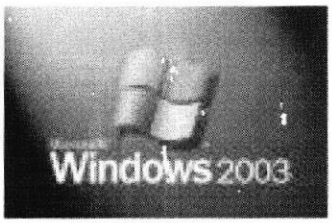


Figura 2.6 Logo Windows 2003[1]

Windows Server 2003 es un sistema operativo de propósitos múltiples capaz de manejar una gran gama de funciones de servidor, en base a sus necesidades, tanto de manera centralizada como distribuida.

2.1.7 Windows XP

Figura 2.7 Logo Windows XP[2]

1 Windows is a registered trademark of Microsoft Corporation in the United States and other countriesMicrosoft Windows Server 2003

[2] Windows is a registered trademark of Microsoft Corporation in the United States and other countries

Este sistema operativo hace uso del nuevo «motor de sistema» que Microsoft desarrolló para Windows 2000, por lo tanto integra altas prestaciones gráficas junto a características de trabajo corporativo (en redes) heredadas de Win2000.

2.2 Comparación a nivel administrativo

2.2.1 Esquema Licenciamiento

Cuando se habla de esquema de licenciamiento se hace referencia al acuerdo que se «firma» entre las dos partes: productor y usuario del software, y que por lo general el segundo descuida (la mayoría de las veces se limita a aceptar el acuerdo sin siquiera leerlo brevemente) pero que constituye un salvamento para el primero.

Se encuentra muy ligado al precio (a pesar de ser dos conceptos diferentes) debido a que por lo general lo que el usuario compra son «licencias», los esquemas de licenciamiento son muy diversos y por lo general varían sustancialmente entre cada compañía productora.

Antes de entrar en materia con la evaluación es importante aclarar que los precios que se incluyen en esta muestra pueden variar según el mercado e incluso dependiendo de la versión o fecha de lanzamiento del sistema. A continuación se dará un vistazo a los esquemas de licenciamiento de cada uno de nuestros sistemas objetos de estudio.

2.2.1.1 FreeBSD

Como ya se dijo inicialmente, este sistema operativo es libre, su *kernel* está licenciado bajo la licencia BSD al igual que gran cantidad de sus aplicaciones, en especial aquellas de integración con el sistema o configuración, sin embargo como la mayoría de UNIX utiliza algunas de las herramientas generadas por el proyecto GNU, por lo cual algunas porciones de FreeBSD están licenciadas por la GPL.

En cuanto al precio, FreeBSD se puede conseguir de forma gratuita descargándolo desde Internet, sin embargo también es posible (ninguna de las dos licencias lo impide) encontrarlo por algún costo, por lo regular bastante bajo comparado a otros sistemas operativos.

2.2.1.2 GNU/Linux

El sistema operativo GNU/Linux se encuentra liberado y protegido a su vez por la licencia pública general o GPL, la cual se destaca por ofrecer por ofrecer las siguientes libertades básicas:

- Libertad para ejecutar el programa, con cualquier propósito.
- Libertad para modificar el programa y adaptarlo a sus necesidades. (Para que esta libertad sea efectiva en la práctica, es necesario disponer del código fuente, porque modificar un programa sin disponer del código fuente es extraordinariamente difícil.)
- Libertad para redistribuir copias, tanto gratis como por un canon.

- Libertad para distribuir versiones modificadas del programa, de tal manera que la comunidad pueda beneficiarse con sus mejoras.

En cuanto al precio, GNU/Linux se encuentra disponible en Internet sin costo alguno, sin embargo incluso los propios fabricantes de las diferentes distribuciones o cualquier persona o empresa puede cobrar por el sistema, los precios pueden ser tan dispares como por ejemplo: USD$ 18 por el conjunto de 7 CDs de Debian 3.0 r1, USD$ 69 por el powerpack de Mandrake 9.2 (7 CDS, 1 manual), USD$ 200 el ProSuite de Mandrake 9.2 (9 CDs, 1 DVD, 2 manuales, incluyendo también la versión para este Sistema Operativo de la base de datos comercial IBM DB2 8.1, además de soporte telefónico), o incluso USD$ 750 por el SUSE LINUX Enterprise Server (cabe aclarar que también incluye software propietario y soporte).

2.2.1.3 Mac OS X

El licenciamiento en el Mac OS X merece especial atención, ya que este sistema operativo incluye un esquema tanto libre como propietario, este sistema tiene su base fundamental en el sistema libre conocido como Darwin, el mismo que es liberado bajo la licencia Apple Public Source License, la misma que en su versión 2.0 ha sido aprobada por Free Software Foundation (Fundación de Software gratis) http://www.fsf.org/fsf/fsf.es.html como una licencia de software libre.

Sin embargo como se mencionó ya, también tiene componentes propietarios, como por ejemplo: la interfaz gráfica Aqua (introducida originalmente con la aparición de los iMac), por lo cual su esquema de licencia también es propietario, el Mac OS X Phanter (la última versión hasta este momento) tiene dos alternativas de compra, 1 usuario (es decir sólo se puede instalar en un computador) por USD$ 129, y el family pack para 5 usuarios por USD$ 199.

2.2.1.4 NetBSD

Este sistema operativo por ser otra variante de BSD y derivado directo de 4.3 BSD se encuentra liberado bajo la licencia BSD, esto implica obviamente que se garantiza su libre redistribución para cualquier fin, bien sea de forma gratuita o mediante el cobro de un canon.

2.2.1.5 OpenBSD

Esta es la tercera variante directa (por que Darwin es indirecta) de los BSD, al igual que los anteriores es software libre y se rige por la licencia BSD, la cual garantiza su libre redistribución, aunque sujeto a los siguientes lineamientos generales:

- La redistribución debe mantener cualquier nota de Copyright del original, preservando así la propiedad intelectual.
- El software regido por esta licencia se entrega sin ningún tipo de garantía.

2.2.1.6 Windows 2003

Por pertenecer a la misma familia de sistemas operativos el esquema de licenciamiento es similar al de Win98, aunque para el caso de esta versión es aún un poco más complicado, ya que no sólo se debe tener en cuenta el número de máquinas a licenciar, sino también el número de procesadores en cada máquina, en especial si pasa de 2, ya que la licencia (y el mismo programa) deben ser «versiones especiales». Windows 2003 es ofrecido en 4 diferentes versiones:

- MICROSOFT WINDOWS SERVER 2003 STANDARD EDITION
- MICROSOFT WINDOWS SERVER 2003 ENTERPRISE EDITION
- MICROSOFT WINDOWS SERVER 2003 DATACENTER EDITION
- MICROSOFT WINDOWS SERVER 2003 WEB EDITION

Cada una de estas posee características que la hacen más adecuada a determinadas necesidades, la siguiente tabla resume el propósito general de la versión y su precio.

Versión	Descripción	Precio Unitario
STANDARD EDITION	El sistema operativo servidor fiable ideal para satisfacer las necesidades diarias de empresas de todos los tamaños, proporcionando la solución óptima para compartir archivos e impresoras, conectividad segura a Internet, implementación centralizada de aplicaciones y un entorno de trabajo que conecta eficazmente a empleados, socios y clientes. Soporta hasta 4 procesadores y 4 Gb de Memoria RAM.	USD$ 560
ENTERPRISE EDITION	La plataforma preferida tanto por las grandes compañías como por las de tamaño medio para implementar aplicaciones de forma segura, así como servicios Web. Integrándose en infraestructuras apoyando fiabilidad, mejores rendimientos y un elevado valor empresarial, se presenta tanto en 32 como en 64 bit. Soporta hasta 8 procesadores, hasta 64 Gb de memoria RAM y permite clustering de hasta 8 nodos.	USD$ 1275

Tabla 2.1 Precios Windows 2003 Fuente Compusariato

2.2.1.7 Windows XP

Al igual que su antecesor, el Sistema Operativo Windows XP viene en dos versiones y obviamente con dos precios diferentes estas dos versiones son las siguientes:

Home Edition: Es una versión orientada al sector familiar, su precio es de USD$ 135.

Professional: Esta incluye mayores capacidades de seguridad y está orientada al sector empresarial, debido a esto su precio es de USD$195.

Aunque sobra la aclaración, es importante hacer gran énfasis en que los sistemas operativos propietarios, prohíben de forma categórica la copia, redistribución e instalación de los mismos sin que medie un contrato de licencia, por lo tanto cuando alguna persona nos convide a utilizar algún programa propietario sin el previo pago de la licencia y sin que le entreguen algún tipo de certificación de la legalidad de la compra se debe pensar muy bien y ser totalmente conscientes que se está cometiendo un acto ilegal en la mayoría de países del mundo, e incluso estamos ante un problema ético.

Por esta razón y aunque suene un poco reiterativo, se exhorta a abandonar estas prácticas que dejan mucho que desear de un profesional, y recomienda siempre actuar dentro del marco de la legalidad, obviamente si no desea verse atado a este tipo de contratos que en muchas ocasiones impiden incluso buenas prácticas como la ayuda a los amigos, se recomienda la utilización de software libre.

2.2.2 Estabilidad y Desempeño

Este tópico es bastante crítico y controvertido, ya que a pesar que existen varias pruebas para medir el rendimiento de un sistema operativo, estas se desempeñan en condiciones «generadas». Por lo general son conducidas en laboratorios y con máquinas especialmente puestas a punto para la comparación, por lo tanto, no son muy fiables cuando se trata de evidenciar la realidad del uso diario.

Cuando se habla de estabilidad se hace referencia a las capacidades de la máquina de soportar la carga sin tener alteraciones que impliquen bloqueo o reinicio, lo cual se deriva en tiempo perdido e incluso posible perdida de la información. El desempeño se refiere a la relación trabajo/tiempo, es decir, a la mejor administración de los recursos del sistema logrando una ejecución rápida de las aplicaciones pero sin descuidar la estabilidad.

A continuación hablaremos un poco acerca de esta característica en los sistemas operativos que estamos evaluando.

2.2.2.1 FreeBSD

Sin duda alguna FreeBSD es uno de los sistemas operativos más estables del mercado, obviamente esto sucede en gran parte por ser una variante de UNIX, aunque va un poco más allá.

Por ejemplo, en las estadísticas de servidores web que realiza la consultora independiente Netcraft http://www.netcraft.com, los 50 sitios web que más tiempo han durado sin tener que reiniciarse están repartidos casi equitativamente entre máquinas con sistema operativo FreeBSD y OpenBSD.

En cuanto a rendimiento también podemos observar que una buena cantidad de sitios en Internet se encuentran soportados por este sistema operativo.

2.2.2.2 GNU/Linux

Aunque el rendimiento puede variar entre una distribución y otra y depende en gran medida del hardware sobre el que se está ejecutando, GNU/Linux en un sistema operativo bastante confiable tanto en rendimiento como en estabilidad.

Debido a su gran flexibilidad es posible «personalizarlo» de múltiples maneras, de tal forma que cumpla con los requerimientos necesarios, no es necesario que se tenga una interfaz gráfica, ya que va a consumir recursos de memoria y procesador que pueden en algún momento necesitar los servidores.

En cuanto a estabilidad como buen derivado de UNIX, GNU/Linux es altamente estable, a manera de comentario y/o evaluación personal, en los cerca de 4 años de trabajo con este sistema operativo no he obtenido más allá de 10 o 20 bloqueos de la máquina, la mayoría más a causa de algún tipo de «experimento» que por inestabilidad del sistema.

2.2.2.3 Mac OS X

La estabilidad en los sistemas Macintosh ha sido altamente calificada a través de su historia, ahora teniendo como base otro gran sistema operativo en términos de estabilidad, Apple ha logrado una muy buena combinación y este nuevo producto se considera ahora

suficiente competencia incluso a sistemas como GNU/Linux o FreeBSD en términos de confiabilidad.

Así mismo el trabajo a nivel de rendimiento en esta plataforma parece (según algunos usuarios) superar a las versiones anteriores de este mismo sistema operativo.

2.2.2.4 NetBSD

El desarrollo de NetBSD se caracteriza por su «limpieza» en la codificación para permitir la portabilidad, sin embargo esto a su vez trae otro beneficio añadido como lo es la estabilidad y el desempeño.

La principal meta de NetBSD no es brindar cada vez cantidad de nuevas características que puedan resultar incompatibles entre algunas plataformas, sino que por el contrario procura escribir bien y mantener un código sin mayores problemas para que su mantenimiento sea fácilmente realizable.

2.2.2.5 OpenBSD

Aunque el principal propósito de OpenBSD es la seguridad, esto no implica que su rendimiento y estabilidad no sean los más adecuados, esto se debe en gran parte a que por más seguro que pueda llegar a ser un sistema operativo si no es lo suficientemente estable como para que esa seguridad extra sea utilizable, este no llegaría a ser ni

medianamente utilizado; la estabilidad y rendimiento de FreeBSD, está dada por los sitios expuestos a Internet que se caracterizan por su alto tiempo sin necesidad de mantenimiento.

2.2.2.6 Windows 2003

Este es uno de los mejores sistemas Windows que han sido desarrollados, está construido sobre la robustez y fiabilidad de Microsoft Windows 2000 Server. Su estabilidad puede llegar a compararse con la de algunos UNIX, hecho que le ha permitido incluso ganar una porción del mercado de los servidores en Internet, es el Sistema Operativo Windows más rápido, fiable y seguro que jamás haya existido.

2.2.2.7 Windows XP

En sus primeras versiones realmente tenía varias deficiencias en cuanto a estabilidad, sin embargo en la actualidad (después del Service Pack 4) ha logrado mejorar ampliamente en este aspecto aunque sin lograr superar a Win 2003.

2.3 Comparación a nivel técnico

En la sección anterior se pudieron apreciar algunos detalles que diferencian o relacionan a los sistemas operativos de nuestra evaluación pero sin adentrarnos en detalles que también son altamente importantes para una persona encargada de la

administración de los recursos informáticos en una empresa pero que al usuario promedio no le son de gran relevancia, en esta sección se pretenden abordar algunos de estos detalles para poder formarnos una opinión más completa de cada sistema y poder de esta forma identificar cual (o cuales) son los que mejor se adaptan a nuestras necesidades.

2.3.1 Compatibilidad con Otras Plataformas

Cuando se habla de compatibilidad con otras plataformas se deben tener en cuenta diferentes conceptos:

- **Soporte a Sistemas de Archivos diferentes al nativo**: Este concepto se asocia a la capacidad que tiene una plataforma de leer y/o escribir en sistemas de archivos de otras plataformas diferentes.
- **Ejecución de Aplicaciones compiladas para otras plataformas**: Algunos sistemas operativos son capaces de ejecutar aplicaciones que fueron originalmente diseñadas para trabajar en una plataforma diferente, la mayoría de las veces (en especial cuando son sistemas bastante diferentes) esto se lleva a cabo por intermedio de otra aplicación que actúa como una capa intermedia entre el sistema operativo y la aplicación no nativa, esta capa permite **emular** las llamadas al sistema y algunos otros procesos del sistema operativo para que el programa crea que se está comunicando con la plataforma para la cual fue compilado.

- **Compartir recursos en red con otras plataformas**: No se puede desconocer que el mundo actual es un mundo «conectado», es decir las redes y la comunicación entre diferentes máquinas es algo cada vez más indispensable, por lo tanto el sistema operativo debe ser capaz de comunicarse con sistemas diferentes en otras máquinas, y obviamente brindarles la posibilidad de utilizar sus recursos (por lo general archivos e impresoras, aunque realmente podría ser cualquier dispositivo).

2.3.1.1 FreeBSD, NetBSD, OpenBSD

Estos sistema operativos poseen soporte para lectura y escritura de una buena variedad de sistemas de archivos, entre los que cabe destacar: FAT16 (DOS, todos los Windows antes de Win 98), FAT32 (Win98, WinMe), ext2 y ext3 (GNU/Linux); para sólo lectura se pueden reseñar: NTFS (Win NT, 2000, XP). Poseen compatibilidad binaria para ejecutar aplicaciones de GNU/Linux, BSD y SCO; también es posible ejecutar aplicaciones DOS (a través del emulador DOSEmu) y Windows (con Wine). Para poder compartir recursos con otros UNIX, se utiliza el protocolo NFS (*Network File System*), mientras que para compartir con Windows (protocolo SMB) utiliza Samba.

2.3.1.2 GNU/Linux

GNU/Linux posee soporte entre otros para los siguientes tipos de sistemas de archivos:

- **JFS**: El *Journaled File System* es un sistema de archivos desarrollado por IBM para sus servidores de alto nivel (*enterprise*) y que comparte con la comunidad en su decidido apoyo a este sistema operativo.
- **XFS**: Este sistema de archivos es el utilizado por SGI en su sistema operativo y que ha puesto a disposición de la comunidad.
- **Minix FS**: Fue el primero que soportó Linux, como su nombre lo indica es el sistema de archivos del Minix del profesor Tanenbaum.
- **FAT**: Linux soporta tanto FAT 16 como FAT 32.
- **NTFS**: El sistema de archivos de NT (Win 2000, 2003 y XP) es totalmente soportado para la lectura, para la escritura las nuevas versiones del kernel (2.6.x) ya traen esta opción, aunque advierten que aún esta en etapa experimental.
- **ADFS**: Es el que utiliza el S.O. *RISC OS*, bastante usado en procesadores RISC. En el kernel aún se encuentra en etapa experimental.
- **BeFS**: El File System del S.O. *BeOS* es posible leerlo aunque también se encuentra en etapa experimental.
- **BFS**: Es el *Unixware Boot Filesystem.*
- **EFS**: Es el antiguo sistema de archivos del S.O. *Irix* de SGI.
- **HPFS**: Sistema de Archivos del OS/2

En cuanto a ejecución de aplicaciones de otras plataformas, Linux soporta el formato binario ELF, el cual es utilizado por varios otros

sistemas UNIX, por lo cual es posible correr algunas aplicaciones creadas para estos otros sistemas.

Además en GNU/Linux existen los emuladores DOSEmu y Wine, por lo que es posible ejecutar aplicaciones diseñadas para sistemas Windows.

La compatibilidad en red con otras plataformas se da por medio de los protocolos NFS, SMB (samba), NCP (NetWare Core Protocol) para compartir con Novell NetWare, Coda (un avanzado sistema de archivos para redes utilizado en varios UNIX), InterMezzo (sistema de archivos bastante utilizado en sistemas distribuidos de alta disponibilidad), Andrew File System (otro sistema de archivos distribuido).

2.3.1.3 Mac OS X

Como Darwin está basado en tecnología FreeBSD y Mach, tiene la posibilidad de soportar binarios de otros UNIX, en especial los del tipo BSD y Linux, pero como además tiene componentes Mac puede soportar también aplicaciones hechas para versiones anteriores de ese sistema, en cuanto a ejecución de aplicaciones Windows el Mac OS X lo puede hacer por medio de un programa llamado Virtual PC, el cual debe comprarse de manera independiente.

Para compartir recursos utiliza Samba y NFS.

2.3.1.4 Windows 2003 y XP

Estos Sistemas Operativos soportan aplicaciones para Windows de 32 bits (Win 98 en adelante), las aplicaciones de 16 bits (DOS, Win 3.x, 95) por lo general no se pueden ejecutar. Los sistemas de archivos soportados son FAT 32 y NTFS. La forma de compartir archivos también es por SMB, aunque por medio de un programa vendido por la misma compañía (Services For UNIX -SFU-) es posible compartir por medio de NFS, al igual que Win 98 también se puede compartir con Novell Netware.

2.3.2 Portabilidad

Cuando en sistemas operativos se habla de portabilidad esto hace referencia a la posibilidad de utilizarlos en diferentes tipos de procesadores incluso si tienen arquitecturas o diseños diferentes.

2.3.2.1 FreeBSD, OpenBSD, GNU/Linux

Además de la tradicional plataforma Intel x86 (386, 486, Pentium, PII, PIII, PIV), estos sistemas operativos también se pueden ejecutar en un buen número más de plataformas, entre ellas se puede resaltar: Intel ia64, Sparc, UltraSparc, PPC (Macintosh), Alpha y Alpha64. Cada día es mayor la cantidad de dispositivos portátiles (relojes, PDAs, Equipos de cocina, equipos de entretenimiento -XBOX, Sega, PlayStation, etc-) en los que se puede encontrar como sistema operativo una versión reducida de Linux.

2.3.2.2 MacOS X

Este sistema operativo ha sido diseñado tan solo para la arquitectura típica de los Mac, es decir, los procesadores Power Mac (o Power PC - PPC).

2.3.2.3 NetBSD

En términos de portabilidad este sistema es el vencedor absoluto, como ya se mencionó esta es precisamente su meta principal, encontrándose portado a más de 50 diferentes procesadores que van desde los grandes servidores (Sparc, Alpha), pasando por los de escritorio (x86, PPC), e incluso en dispositivos portátiles (consolas de juegos y otros).

2.3.2.4 Windows

Los sistemas Windows de los que hemos hablado hasta ahora se ejecutan sobre arquitecturas Intel x86, pero existe una versión modificada denominada Windows CE, orientada al mercado móvil.

2.3.3 Requerimientos mínimos de Hardware

En esta sección se resumen los principales requerimientos mínimos de hardware para cada uno de los sistemas que hemos venido estudiando; sin embargo, los sistemas FreeBSD, GNU/Linux, NetBSD y OpenBSD pueden correr en diferentes plataformas de hardware, por lo cual es

lógico que sus requerimientos pueden variar, por este motivo, se ha decidido sólo mostrar los requerimientos en la plataforma x86, excepto como es lógico en el caso del Mac OS X, donde se utilizará el procesador Power PC.

S.O.	Procesador Mínimo	Mínimo de RAM	Espacio mínimo en HD
FREEBSD	386	16 Mb para el programa de instalación, después de esto se puede reducir hasta 4 Mb.	10Mb a 50Mb
GNU/LINUX	386	4 Mb	100 Mb
MAC OS X	PowerPC G3	128 Mb	2 Gb
NETBSD	386	4 Mb	50 Mb
OPENBSD	386	8 Mb	50 Mb
WIN 2003	Pentium 550 Mhz	256 Mb	1.5 Gb
WINDOWS XP	Pentium 233 Mhz	128 Mb	1.5 Gb

Tabla 2.2 Requerimientos Hardware

Es bueno aclarar que estos requerimientos deben variar bastante dependiendo de las necesidades de la instalación, es decir no se necesitaran los mismos requerimientos (en especial en el caso de los UNIX) si se desea tener sólo una máquina que actué como *router* o como servidor de archivos, para lo cual no es necesario una interfaz gráfica, que si se deseara tenerla para el uso cotidiano (suite de oficina, juegos, etc). Incluso estos requerimientos base pueden llegar a cambiar, existen proyectos (ya funcionando) en los que se recompilan con capacidades mínimas núcleos de sistemas como *BSD y Linux y es posible tener un computador ejecutándose con el sistema operativo en un disquete tradicional. Así mismo como ya se dijo si se desea una interfaz gráfica

los requerimientos de hardware pueden llegar hasta el punto de duplicarse (o algo más).

Esto también aplica para los sistemas Windows, excepto por que en estos no es posible suprimir la interfaz gráfica ni algunos otros componentes.

SO	Conectividad	Confiabilidad	Estabilidad	Escalabilidad	Multi usuario	Multi Platafor ma	POSIX	Prop.
UNÍS	Excelente	Muy Alta	Excelente	Muy Alta	Si	Si Múltiple	Si	Si
Windows NT	Muy Buena	Baja	Regular	Media	Inseguro	Parcial	Limitada	Si
Netware	Excelente	Alta	Excelente	Alta	Si	Si	No	Si
Linux	Excelente	Muy Alta	Excelente	Muy Alta	Si	Si Múltiple	Si	No

Tabla 2.3 Comparación de las Características Generales de los Sistemas Operativos

2.4 Nivel de seguridad

El medir el nivel de seguridad de un paquete de software no es una tarea sencilla, en él intervienen infinidad de parámetros, los mismos que deben ser evaluados con una capacidad de variación muy grande sobre la evaluación final.

La metodología con la que se intentará evaluar la calidad de software producido en un modelo de desarrollo de Software Libre se basará en su comparación con paquetes de software cerrado/propietario/privativo (desde ahora nos referiremos a este modelo de desarrollo de software como Software Cerrado).

De esta forma, este trabajo lejos de intentar cuantificar la calidad de cada uno de los modelos de software con una valoración final para cada uno de ellos, los comparará entre sí con el fin de encontrar el mejor de los dos.

Por supuesto, el resultado y la valoración final de este, habrá que acotarla a los ámbitos en los que se ha centrado el estudio, es decir, ni se han expuesto todos los modelos de desarrollo de software, ni se han comparado todos los paquetes de software de cada uno de los dos modelos estudiados.

También hay que tener en cuenta que en los últimos años, está creciendo el número de programas de Software Cerrado que incorporan en su interior fragmentos de código y librerías libres - normalmente licenciadas bajo BSD o LGPL. Esto hace un poco más difícil aún el poder comparar los dos métodos de desarrollo.

El Software Libre es un movimiento de colaboración global en el que se desarrollan programas de una forma abierta. El código fuente de estos programas está disponible para que cualquier persona lo descargue, lea, use, modifique o redistribuya. El desarrollo se basa en la comunicación por medio de Internet, y se realiza entre grupos de gente que en muchas ocasiones no se conocen en persona.

Al contrario que en el desarrollo de Software Cerrado, en el Software Libre no se sigue ninguna especificación de desarrollo basada en la Ingeniería del Software: El no seguir un conjunto de normas estrictas hace que se

genere un software más inseguro. Por otro lado, en el desarrollo de Software Libre se van liberando versiones de un programa durante su desarrollo.

En muchos casos estas versiones beta de programas se utilizan. Por ejemplo, en el caso del modelo de desarrollo del Software Cerrado, es conocido que con frecuencia existen problemas de falta de tiempo en el desarrollo: una práctica común al contratar un servicio de esta clase es firmar una fecha máxima de finalización del proyecto; en otras ocasiones, son los aspectos relacionados con el marketing y promoción del producto los que requieren que se termine de desarrollar antes de una fecha marcada.

Desde este punto de vista, es posible que haya Software Cerrado en estado -beta- a causa de estas presiones temporales. Dada esta circunstancia, ¿Hay alguna diferencia entre la seguridad de un de programa generado por el modelo de Software Cerrado, con una versión no terminada de Software Libre?

Ahora bien, todos estos pequeños factores no son demasiado relevantes en comparación de, por ejemplo, la motivación que lleva a desarrollar el software.

En el caso del modelo de Software Cerrado, la motivación es puramente económica: Se desarrolla un paquete de software a cambio de una

compensación económica. Un desarrollador de Software Libre puede realizar esa tarea por distintas motivaciones, mayoritariamente éticas o económicas. Si es por razones éticas, comparte el punto de vista más científico de que el software es conocimiento, y como tal conocimiento no debería ocultarse, sino difundirse para progresar.

Si por el contrario lo hace por razones económicas, es porque cree en que es un nuevo modelo de desarrollo de software sostenible y rentable, con el que puede obtener un mejor resultado que con el modelo Cerrado, y en algún caso situarse en un mercado en el que de otra forma sería mucho más difícil acceder.

En cierta forma, al publicarse el trabajo de un desarrollador - grupo, o empresa - de Software Libre se está evaluando en público su calidad.

Al dejar el código fuente accesible, permite que cualquier persona pueda evaluar su calidad, y por lo tanto, su calidad como empresa o profesional.

Además de todos estos factores, existen otros de igual importancia: socialmente hay una gran permisividad con la copia ilegal de software, la adopción de soluciones basadas en Software Libre son relativamente recientes en algunos nichos, etc.

Según los datos de BSA - Business Software Alliance – existen países en los que se copia ilegalmente hasta el 94% del software que se utiliza.

Esta cultura de copiar ilegalmente Software Cerrado también puede estar relacionada con el nivel de seguridad que de él se desprende: Hoy en día, existe un grupo indeterminado de gente y empresas que utilizan Software Cerrado ilegalmente; gracias a este hecho, la base instalada de estos productos - y por tanto, de conocimiento a su alrededor - es mayor de lo que sería sin la existencia de la copia ilegal de Software Cerrado. Tiene esto alguna relación con la calidad del software?. Una mayor difusión y por tanto, una mayor base de conocimientos sobre los productos cerrados hacen suponer en un primer momento que deberían ser más seguros.

País	2004
Paraguay	83%
Bolivia	80%
El Salvador	80%
Nicaragua	80%
Venezuela	79%
Guatemala	78%
República Dominicana	77%
Argentina	75%
Honduras	75%
Perú	73%
Uruguay	71%
Ecuador	70%
Panamá	70%
Costa Rica	67%
México	65%
Chile	64%
Brasil	64%
Colombia	55%

Tabla 2.4 Copia ilegal Latinoamérica Fuente: Business Security Alliance (BSA)

2.4.1 Vulnerabilidad

En los últimos años se han revelado múltiples vulnerabilidades tanto por el lado del Software Libre en productos como: GNU/Linux o FreeBSD,

como en el del Software Cerrado con productos como: Microsoft Windows o Cisco IOS. La ventaja de esta comparación en especial es que se trata de un elemento imprescindible para cualquier dispositivo; gracias a esto, los datos tomados serán mucho más representativos que en otras posibles comparaciones.

En la actualidad existen organizaciones dedicadas a la seguridad informática en diferentes ámbitos y al reporte de vulnerabilidades descubiertas en paquetes software. En concreto, a continuación se presentarán las estadísticas obtenidas de los datos de una de ellas, BugTraq. Durante años, esta organización ha publicado los problemas de seguridad de todo tipo de software. La tabla que se expondrá a continuación muestra el número de vulnerabilidades de los sistemas operativos más extendidos hoy en día.

Sistema Operativo	1997	1998	1999	2000	Total
Windows NT	4	6	99	34	143
Windows 9x	1	1	46	11	59
Solares	24	31	34	6	95
IRIS	26	13	8	3	50
HP-UX	8	5	7	3	23
MacOS X	0	1	6	0	7
GNU/Linux	10	23	84	30	147
Debian	2	2	29	5	38
Red Hat	5	10	38	17	70
FreeBSD	4	2	18	6	30
OpenBSD	1	2	4	2	9

Tabla 2.5 Estadísticas sobre las vulnerabilidades Fuente: Lista BUgtraq

Conviene aclarar cómo hay que interpretar los datos anteriores. La primera parte de la tabla contiene los sistemas operativos de Software

Cerrado, y la segunda parte los de Software Libre. En esta última parte parece una fila con el título GNU/Linux, eso es porque en esta entrada se contabilizan todos los problemas de seguridad de GNU/Linux; se trata de la suma de todas las vulnerabilidades diferentes aparecidas en todas las distribuciones, pero sin duplicar su conteo si aparece dos veces el mismo problema en dos distribuciones diferentes.

Por otro lado hay que estudiar las diferencias entre cada uno de los sistemas operativos:

Los sistemas operativos cerrados tipo UNIX en muchas ocasiones ejecutan algunos programas libres. Típicamente sendmail o bind.

El tamaño de los sistemas operativos es un factor crítico en esta comparación. En estas estadísticas, se ha contabilizado las vulnerabilidades de los sistemas operativos según se distribuyen. Esto quiere decir que se ha evaluado a los sistema libres con varios miles de aplicaciones y a los cerrados únicamente con algunas decenas. El modelo económico del Software Cerrado se basa en la venta de aplicaciones que se ejecutaran sobre un sistema operativo, pero no en la distribución conjunta de estas con el propio sistema.

Existen infinidad de aplicaciones que resultan básicas para el uso cotidiano de un ordenador, pero que en muchos sistemas operativos no se distribuyen por defecto: Herramientas de ofimática, lectores de correo

electrónico, herramientas de trabajo con gráficos, etc. Todas las vulnerabilidades de este software no han sido contabilizadas en las estadísticas expuestas anteriormente. Por ejemplo, las incidencias de seguridad en Microsoft Exchange no fueron contadas ya que se entiende que este software no es parte del sistema operativo. Por el contrario, las incidencias de sendmail si fueron contabilizadas, ya que este servidor si que se distribuye conjuntamente con el resto de los sistemas operativos libres.

Además de la variedad de aplicaciones incluido en una y otra clase de sistema operativo, hay que tener en cuenta que en la mayoría de los casos los sistemas operativos libres no incluyen una única opción para cada una de las herramientas. Por ejemplo, si se trata de un gestor de bases de datos normalmente incluirá MySQL y PostgreSQL, como entornos de escritorio GNOME o KDE, como navegador web Mozilla o Konqueror, como cliente de correo Kmail, Mutt o Evolution, etc.

También hay que tener en cuenta otros muchos factores:

Algunas de las vulnerabilidades son mucho más importantes que otras, no todas son explotadas por los atacantes de la misma forma, la velocidad en el que el desarrollador/fabricante del producto corrige y publica una solución, la velocidad con la que los administradores de sistemas o usuarios lo aplican, etc. Por esta razón, hay que tener en cuenta la forma en la que se trabaja con los reportes de vulnerabilidades

en cada uno de los casos. En los programas libres, al estar todo su código fuente disponible para realizar una auditoria es muy fácil estudiarlo en busca de algún problema de seguridad. Si se descubre un problema, este se hace público - por lo tanto, se contabiliza como tal - y a continuación se corrige y se libera una nueva versión del software. En el caso de software cerrado, si se realiza una auditoria de seguridad sobre su código, los posibles fallos y vulnerabilidades que se detecten serán corregidos, pero en muchos casos no se harán públicas - y por tanto, no serán contabilizados a no ser que alguien detecte el error por medio de la experimentación o el uso de dicho software. Esta forma de auditar el software claramente favorece un menor número de vulnerabilidades públicas en el software cerrado.

La forma de distribuir las correcciones del software también suele ser muy diferente entre los sistemas operativos abiertos y cerrados.

En el caso de sistemas operativos abiertos se tiende a publicar parches enfocados a la corrección de un único error. Estos parches se publican, y a continuación, se libera una nueva versión del programa, y posiblemente una nueva versión de los binarios de los paquetes para su distribución.

En el caso del software cerrado, las compañías tienden a recolectar un conjunto de parches y distribuirlos conjuntamente – algunos fabricantes, como Microsoft, los llaman Service Pack.

Históricamente se ha podido comprobar que hay veces que estos paquetes de parches chocan con parches individuales o que incluso han incorporado nuevos problemas al arreglar los anteriores, esto por lo general es un problema para el usuario.

Se ha generalizado esta situación, de tal forma que llegado al momento en el que la instalación de un navegador Web en un sistema operativo cerrado haya cambiado el kernel del sistema sin previo aviso.

2.4.2 Intrusiones

En la undécima conferencia anual de RSA se expusieron los datos sobre las intrusiones de seguridad durante ese año. En "Security in 2002 worse than 2001, exec says" se exponen los siguientes datos sobre el número de ataques que recibieron cada uno de los sistemas operativos.

Sistema Operativo	Nº ataques
Windows	31 millones
Unix	22 millones
CISCO IOS	7 millones

Tabla 2.6 Conferencia RSA Febrero 2005: Datos sobre intrusiones

En estos datos todos los sistemas operativos Unix aparecen reflejados en el mismo valor. Entre ellos se encuentran tanto los sistemas libre: GNU/Linux y BSD, como los de cerrados: Solaris, HP-UX, AIX, etc. Este hecho invalida en parte estos datos para los propósitos de comparar la seguridad del Software Libre y Cerrado ya que no es posible distinguir

entre los ataques que sufrieron los sistemas libres. En el peor de los casos, y contabilizando todos los Unix como sistemas libres - lo cual no es cierto - los sistemas operativos cerrados hubiesen recibido 38 millones de ataques frente a los 22 millones de los sistemas libres.

2.4.3 Uso en servidores web

Por otro lado existen datos que pueden ayudar a clarificar los anteriores. Netcraft, una empresa dedicada a la estadística y la seguridad en Internet ha publicado los datos sobre los sistemas operativos de los servidores de Internet en ese mismo periodo de tiempo.

Sistema Operativo	Nº servidores	%
GNU/Linux	6,116,811	35.73%
Windows	3,644,187	21.32%
Otros	3,802,268	21.24%
Solaris	3,484,135	20.35%
Desconocido	233,676	1.36%

Tabla 2.7 NETCRAFT 2005: Sistemas Operativos de Servidores de Internet

Con estos datos, y suponiendo un total de 9 millones de servidores ejecutando un sistema operativo libre, la media sería de 2,4 intrusiones por cada sistema. En el caso de los sistemas operativos propietarios la media de intrusiones supera el doble: 5,3.

Este no sería un dato significativo si los sistemas operativos libres fuesen minoritarios y no gozasen de una amplia base instalada, pero

basándonos en las estadísticas de NetCraft se puede ver como no es así y, al menos, hay igual número de unos y otros.

2.4.4 Tiempo de reacción

Otro punto de interés respecto a la seguridad del Software Libre y el Cerrado es el tiempo que transcurre entre que una vulnerabilidad es descubierta, hasta que se publica una solución para el producto afectado. Es decir, la cantidad de tiempo que el sistema es abiertamente vulnerable - hasta ese momento lo había sido de igual forma, pero el problema no era público.

Existen estudios sobre la velocidad en que diferentes compañías arreglan y publican soluciones para sus productos. Por ejemplo, SecurityPortal realizó un estudio sobre Microsoft, Sun y RedHat.

Compañía	Intervalo vulnerable	Vulnerab.	Media
Red Hat	348	31	11,23
Microsoft	982	61	16,10
Sun	716	8	89,50

Tabla 2.8 SECURITYPORTAL 2005: Velocidad de reacción.

En dicho estudio se puede ver como la compañía que tuvo un menor intervalo de vulnerabilidad fue Red Hat; esto quiere decir, a igualdad de atención en la actualización de un sistema Windows y uno GNU/Linux de

Red Hat, el sistema no libre fue aproximadamente tres veces más vulnerable que el libre.

Un comentario especial merece el caso del Solaris, el sistema operativo de Sun, ya que fue el que sufrió un menor número de vulnerabilidades, pero fue el que más tardó en arreglarlas, haciendo que su resultado final fuese muy inferior a la de las otras compañías.

Por último hay que tener en cuenta que, en el caso de Red Hat, pueden transcurrir días desde que una vulnerabilidad de un programa libre es corregida hasta el momento en que las empresas dedicadas a las distribuciones adoptan la solución, reconstruyen sus paquetes y los ponen disponibles para sus usuarios; es decir, normalmente es posible disponer de una versión corregida del software antes de que la propia empresa publique su versión.

Esto reduciría aun más el tiempo en el que un programa libre se encuentra comprometido por una vulnerabilidad.

Los datos anteriormente expuestos pueden tener más o menos relevancia en función de cuán importante es que un sistema se encuentre vulnerable. Si existe un bajo riesgo de que la vulnerabilidad sea atacada y/o las consecuencias del ataque son pequeñas, entonces no tendrán gran importancia. Si por el contrario, las consecuencias son más severas y/o existe un gran riesgo de que el sistema sea atacado, el

tiempo en el que el sistema es vulnerable se vuelve fundamental para su seguridad. A este respecto, el CERT ha publicado un estudio: consiste en la instalación de un sistema vulnerable y la posterior toma de datos para, de esta forma, ver cuanto tiempo tardaba en ser comprometido y de qué ataques era objeto.

Los datos fueron los siguientes:

- A las 8 horas, se detectaron pruebas de vulnerabilidades en el RPC
- A los 21 días, se habían detectado 20 intentos de ataques conocidos
- A los 40 días, el sistema estaba comprometido por medio de una vulnerabilidad en el servidor POP3.
- El intruso instaló un sniffer, varias puertas traseras y modificó los logs del sistema.

Con estos datos, ya es posible valorar adecuadamente los del estudio de Security Portal, y comprobar como el sistema basado en Software Libre es efectivamente mucho más seguro que el sistema basado en Software Cerrado.

2.4.5 Virus, gusanos y troyanos

Los virus, gusanos o troyanos son también un aspecto a tener en cuenta en esta evaluación. En esta ocasión, este es un aspecto derivado de la calidad del software producido por cada modelo: si el software tiene

alguna clase de deficiencia, es posible que un gusano la explote. Una buena prueba para comprobar el impacto de esta clase de ataques la constituye la lista de trabajos actuales del US-CERT[1].

Actualmente (Fri, May 27 2005) la lista de incidencias esta compuesta de:

- W32/Sober
- Algunas variantes del código malicioso W32/MyDoom
- Exploit for Microsoft PCT vulnerability released
- Gusano MySQL UDF
- W32/Zafi.D
- W32/Bagle

Todas ellas son vulnerabilidades de programas desarrollados con el modelo cerrado, aunque caben algunos posibles argumentos para este hecho.

Seis de las nueve incidencias de seguridad son virus de Microsoft Windows; es posible que este hecho se deba a que existen muchos más desktops ejecutando este sistema operativo cerrado.

[1] United States Computer Emergency Readiness Team

Parece lógico pensar que los escritores de esta clase de programas se centrarán en la plataforma que domine el mercado, para que sus creaciones tengan la mayor repercusión posible.

También sería posible añadir los hechos anteriormente expuestos: si el software cerrado es más vulnerable por su forma de desarrollo y/o su mantenimiento, es lógico que existan más programas maliciosos que se aprovechen de esta deficiencia.

2.4.6 Servidores Web

Por último, el software de los servidores web serán la última herramienta que se estudiará para comparar la seguridad del Software Libre y Cerrado.

Servidores Web	Número	Porcentaje
Apache	33,892,817	67,05%
IIS	10,858,168	21,48%
SunOne	1,644,412	3,25%
Zeus	763,302	1,49%

Tabla 2.9 NETCRAFT: Servidores web, Mayo 2005

Ahora habría que estudiar el número de incidencias que se producen en cada uno de ellos.

Si se cruzan estos últimos datos con los de popularidad en servidores de Internet es posible obtener un valor sobre el porcentaje de incidencias.

Hay que tener en cuenta qué este valor no es completamente justo, ya que los sistemas con una base instalada más amplia tienen una posibilidad mayor de sufrir ataques: un atacante tiene una posibilidad mayor de encontrar uno mal configurado o vulnerable, existe un background mayor sobre la tecnología, etc.

2.4.7 Ataques exitosos a Servidores Web

Un estudio de Attrition a este respecto muestra los porcentajes de ataque que sufrieron los servidores ordenados por el sistema operativo que ejecutaban.

Sistema Operativo	Porcentaje
Windows	66,09%
GNU/Linux	17,01%
Solaris	8%
*BSD	6%

Tabla 2.10 ATTRITION: Ataques exitosos a servidores web, 2005

De los datos del estudio de Attrition de sacan varias conclusiones. En primer lugar se ve claramente que Microsoft Windows es el sistema más vulnerable a los ataques Web. En lo relativo a los sistemas operativos libres, se puede observar como GNU/Linux sufre más ataques exitosos que la familia de los BSD; siendo todos ellos libres, parece que existe un mayor nivel de seguridad en unos que en otros.

Arthur Wong, CEO de SecurityFocus anunció en "RSA: Security in 2002 worse than 2001, exec says" el resultado de las estadísticas de su compañía respecto a la seguridad de los servidores Web. Los datos

presentan que IIS[1] fue atacado 1400 veces más que Apache: mientras que el servidor de Microsoft había sido atacado 17 millones de veces, Apache lo había sido 12,000. Estos números tienen una especial relevancia si se tiene en cuenta que hay aproximadamente tres veces más servidores Apache que IIS en Internet. El número total de ataques de toda clase de sistemas en 2002 fueron 29 millones.

2.5 Migración de Amazon.com a Linux

Con su cambio al uso de Linux para manejar ciertas operaciones claves, Amazon logró ahorrar $17 millones en el pasado trimestre, una cantidad de dinero crítica en este trimestre. Amazon se une a United y Korea Air Lines, entre otras multinacionales, en su uso generalizado a nivel corporativo de Linux, una versión modificada de Unix que presenta claras ventajas:

- Es gratis, para una computadora o para miles (no hay que comprar licencias.)
- Se puede hacer "upgrades" sin costo alguno en la mayoría de los casos.
- Es un sistema operativo más simple y más robusto que Unix o Windows.

[1] Internet Information Services

El cambio de Amazon obedece a su necesidad de mantener sus sistemas operando cada segundo de cada día y la combinación de ahorros/eficacia/expansión que Linux ofrece.

Según Microsoft, cambiar a Linux tiene "costos escondidos", especificamente que obliga al cliente a convertirse en manejador del sistema operativo en términos de mantenimiento y adelantos. Microsoft alega que pequeñas y medianas empresas no pueden absorber este costo.

Pero un estudio realizado por el consultor independiente Rob Valliere indica que para empresas pequeñas (menos de 35 empleados), los ahorros por el uso de Linux pueden ser de $5,000 a $24,000 al año, y que el único producto de Microsoft que excede en eficacia lo que existe actualmente de Linux es Outlook (que es el mejor programa jamás creado por Microsoft y uno de los 10 programas más importantes de la Era del Internet.) Y para empresas que comienzan, una instalación completa (servidor y software) de Linux para 35 personas promedia $6,300, mientras que una instalación similar de Windows 2000, con Office 2000 (35 copias de cada uno) promedia $27,000.

Y por último, en otro estudio publicado por IT News, se reportó que el promedio de tiempo de apoyo técnico para Windows NT instalado en empresas es de 37 horas al año. ¿Para Linux? 26 minutos. Al año.

2.6 Conclusión

No es sencillo extraer una única conclusión de todo lo expuesto anteriormente. A lo largo del capitulo se han tratado los posibles factores que intervienen en la mayor o menor seguridad del Software Libre respecto al Software Cerrado.

Basándose en los datos expuestos, los sistemas basados en Software Libre son claramente más seguros. Las razones para este fenómeno podrían ser varias: es posible que sea por la motivación con la que se desarrolla, por la forma de trabajo, por la gestión de las vulnerabilidades o incluso por algunos aspectos sociales.

Tampoco es seguro que esta situación se vaya a perpetuar, es decir, no hay una razón clara por la que esta situación no pudiera cambiar en un futuro, aunque tampoco hay ningún motivo para suponer que se producirá tal cambio. El hecho es que, hoy en día, los sistemas basados en Software Libre tienen un menor porcentaje de ataques con éxito, su actualización frente a vulnerabilidades es más ágil, están ampliamente implantados en muchos sectores y su tendencia de todo este movimiento es el seguir creciendo.

Todos los Sistemas Operativos analizados en el presente capítulo representan opciones viables para la implementación de seguridad en los servidores. Red Hat Linux es un Sistema Operativo que debe considerarse seriamente ya que presenta numerosas ventajas, además de lo económico

de su adquisición, las herramientas de seguridad que incluye hacen factible su configuración como servidor Web.

Los Requerimientos de Hardware para la Instalación de Red Hat son otra ventaja en la utilización de este Software ya que demanda pocos recursos para un funcionamiento óptimo. Por tanto los costos de adquisición de Hardware disminuyen considerablemente en relación a otro Sistema Operativo. Aunque debe verificarse la Lista de Compatibilidad de Hardware previamente a su adquisición.

CAPÍTULO 3

3. SERVIDORES WEB - HERRAMIENTAS DE CONTROL Y SEGUIMIENTO

Es importante que los administradores de sistemas estén enterados de los hackers, de las herramientas que emplean y el software necesario para supervisar y controlar tal actividad. Las siguientes herramientas permiten probar la integridad de los servidores.

En este capítulo se encuentran aquellas herramientas que nos permitirán tener información, mediante archivos de trazas o logísticos, de todos los intentos de conexión que se han producido sobre nuestro sistema o sobre otro que nosotros hayamos señalado, así como intentos de ataque de forma sistemática a puertos tanto de TCP como de UDP.

Algunas de las herramientas descritas en este capítulo no necesitan estar instaladas en la máquina que se quiere controlar, ya que se puede poner en una máquina cuya interfaz de red funcione en modo promiscuo, permitiendo seleccionar la IP o máquina que queremos auditar. Otras herramientas descritas en este capítulo pueden tener un doble uso, nos permiten protegernos ante posibles ataques, pero también podrían ser utilizadas para intentar comprometer los sistemas. Es importante que el uso de estas herramientas esté restringido, para que no todo el mundo

esté utilizándolas de forma aleatoria y nos oculten realmente un ataque. También podrán ser utilizadas para realizar seguimientos en la red cuando creamos que alguna de nuestras máquinas ha sido comprometida.

Las herramientas que permiten este tipo de operatividad son: tcp-wrapper, netlog, argus, tcpdump, SATAN, ISS, courtney, gabriel, nocol, tcplist.

3.1 Tcp-wrappers

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

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

Una vez instalado, se pueden controlar los accesos mediante el uso de reglas y dejar una traza de todos los intentos de conexión tanto admitidos

como rechazados (por servicios, e indicando la máquina que hace el intento de conexión).

El programa utiliza el syslogd (puerto 514 udp) para mandar esa información; por defecto utilizará la salida de mail, la cual estará indicada en el archivo de configuración de syslogd (/etc/syslog.conf) con la línea mail.debug. Esto se puede cambiar en los fuentes del programa y se puede re-dirigir a otro lugar mediante el uso de las variables de usuario que deja libres el syslogd, estas variables vienen definidas en el archivo /usr/include/syslog.h). Una vez modificados los fuentes, se deberá indicar al syslogd donde debe dejar la información de esa variable local.

En referencia al control de conexiones a distintos servicios, el software utiliza dos archivos de información (hosts.allow, hosts.deny) situados en el directorio "/etc". Es en estos archivos donde se definirán las reglas que deben utilizarse para el filtrado de los paquetes. El filtrado se puede realizar teniendo en cuenta tanto a máquinas como a servicios o una mezcla de ambos.

En el caso de las máquinas hay varias formas de hacerlo. Por ejemplo se le puede indicar que sólo se conecten las máquinas pertenecientes al mismo dominio (esto se puede ampliar a los que tengan el mismo rango de direcciones IP, para evitar que máquinas no definidas en el DNS no

puedan conectarse), o sólo aquéllas cuyos nombres sean especificados de forma explícita.

Veremos, a continuación, un ejemplo que consiste en permitir las conexiones sólo de aquellas máquinas de mi dominio. Para ello tendríamos que disponer de lo siguiente:

```
hosts.deny
        ALL: ALL
hosts.allow
        ALL: LOCAL, sfp.gov.ar
```

La secuencia es la siguiente: en el primer archivo denegamos todas las conexiones; mientras que en el segundo, permitimos las conexiones de las máquinas definidas en mi mismo dominio. Una utilidad que puede ser interesante a la hora de tener información de conexiones de forma automática es el uso de comandos en estos archivos. Es decir, podemos decirle al programa que cuando se produzcan ciertas conexiones ejecute un comando.

Veamos un ejemplo:

```
hosts.deny
        ALL: ALL (/usr/ucb/finger -l @%h |
/usr/ucb/mail -s %d%h e-mail) &
hosts.allow
        ALL: LOCAL, uc3m.es
```

Según este ejemplo, cuando se produzca una conexión no deseada, de forma automática se ejecutará un finger a la máquina que origine esa conexión y el resultado del mismo, se mandará vía correo electrónico al usuario especificado (el administrador de la máquina o el responsable de seguridad de la organización), indicando en el "subject" del mensaje el servicio al cual se intento la conexión y la máquina que lo originó, y como cuerpo el resultado del finger sobre esa máquina.

Además de estas reglas podemos incluir protocolos específicos de control, veamos un pequeño ejemplo de esta utilidad:

```
hosts.deny
        ALL: ALL
hosts.allow
        in.ftpd: LOCAL, sfp.gov.ar
```

Según este ejemplo, sólo permitimos conexiones FTP desde nuestro dominio, prohibiendo las demás conexiones que estén filtradas por el programa.

Estos ejemplos son muy básicos, pero el grado de complejidad de las reglas puede aumentar incluyendo distintos protocolos y listas de máquinas por protocolos. Como mencionábamos anteriormente, este tipo de programa genera trazas de las conexiones establecidas. Veremos, a continuación, unas recomendaciones referentes a las trazas que pueden extenderse a otro tipo de utilidades.

Es aconsejable disponer de una o varias máquinas para centralizar las trazas que creamos convenientes. Describiremos ahora una posible organización para tener información de todas las conexiones que se producen en nuestro grupo de máquinas.

Podemos clasificar nuestras máquinas por sistema operativo o por funciones que realizan. A cada uno de estos grupos se le asigna una variable en el syslog (como veíamos anteriormente), y se envía (vía syslog) a una o varias máquinas (cuya finalidad es que tengan todas las conexiones que se produzcan en tiempo real).

Disponer de varias máquinas de este tipo es muy útil ya que los hackers suelen correr programas del tipo "zap", que borran las huellas en el sistema asaltado, y este tipo de herramienta (tcp-wrapper) que deja las trazas en un archivo tipo texto, sería muy fácil su modificación editando el archivo con un editor de texto.

De hecho, esto puede constituir una pista de que un sistema ha sido asaltado. Es decir, que los archivos de trazas relativos a una máquina sean distintos en la máquina que lo originan y en las máquinas que lo centralizan.

Debemos tener en cuenta que las máquinas que centralizan esta información deben estar muy protegidas ante los posibles ataques.

Para concluir, podemos decir que el tcp-wrappers es una simple pero efectiva herramienta para controlar y monitorear la actividad de la red en nuestra máquina, y nos permite un control sobre las conexiones que se efectúan en nuestra red. Veamos un pequeño ejemplo del archivo de trazas que genera este software.

```
May 29 13:21:22 lince.uc3m.es in ftpd[237]: connect from acme.uc3m.es
May 29 13:52:00 lince.uc3m.es in.ftpd[557]: connect from acme.uc3m.es
May 29 13:54:21 lince.uc3m.es in.telnetd[561]: connect from acme.uc3m.es
May 29 14:50:20 lince.uc3m es in.ftpd[8228]: refused connect from
acme.uc3m.es
May 29 14:51:12 lince.uc3m.es in.ftpd[8232]: connect from acme.uc3m.es
May 29 14:57:33 lince.uc3m.es in.ftpd[8275]: connect from acme.uc3m.es
Jul 25 13:47:45 lince.uc3m.es in.telnetd[338]: refused connect from
acme.uc3m.es
Jul 25 13:48:16 lince.uc3m.es in.telnetd[351]: refused connect from
acme.uc3m.es
```

Figura 3.1 Ejemplo del archivo de trazas generado por Tcp-wrappers

3.2 Netlog

Este software de dominio público diseñado por la Universidad de Texas, es una herramienta que genera trazas referentes a servicios basados en IP (TCP, UDP) e ICMP, así como tráfico en la red (los programas pueden ejecutarse en modo promiscuo) que pudiera ser "sospechoso" y que indicara un posible ataque a una máquina (por la naturaleza de ese tráfico).

El paquete está formado por el siguiente conjunto de programas:

- Tcplogger
- Udplogger
- Icmplogger
- Etherscan

3.2.1 Tcplogger

Este programa escucha todos los servicios sobre TCP, dejando una traza de cada servicio en un archivo de trazas, indicando la hora, la máquina origen y el puerto de esa conexión. Veamos un pequeño ejemplo de un archivo originado por este programa:

```
04/25/03    14:23:50  6C0J6A00 arapaima.uc3m.es   1153 -> acme      telnet
04/25/03    14:29:50  4D4C1600 elrond             3865 -> acme      smtp
04/25/03    14:35:30  396f5600 svin09.win.tue.nl  ftp-data -> acme 2527
04/25/03    14:53:26  18DB4800 ordago             3268 -> acme      smtp
04/25/03    14:54:49  58ECA600 elrond             3880 -> acme      smtp
04/25/03    14:59:58  399B7801 siuX               1529 -> acme      telnet
04/25/03    15:27:27  4562B200 sun.rediris.es     1617 -> acme      domain
```

Figura 3.2 Ejemplo del archivo de trazas generado por Tcplogger

3.2.2 Udplogger

Es semejante al anterior, pero para los servicios sobre UDP. Un ejemplo del archivo de trazas:

```
10/23/03    11:25:04   0 d.root-servers.net  domain -> acme     domain
10/23/03    11:25:05   0 elrond              1659 -> acme       domain
10/23/03    11:25:05   0 elron.              1660 -> acme       domain
10/23/03    11:25:05   0 crl.dec.com         domain -> acme     domain
10/23/03    11:25:06   0 acme                4083 -> acme       domain
10/23/03    11:25:06   0 acme                4084 -> acme       domain
10/23/03    11:25:06   0 darkstar.isi.edu    domain -> acme     domain
10/23/03    11:25:06   0 acme                4087 -> acme       domain
```

Figura 3.3 Ejemplo del archivo de trazas generado por Udplogger

3.2.3 Icmplogger

Se encarga de trazar el tráfico de ICMP. Veamos un ejemplo del archivo de trazas:

```
10/23/03    11:24:08    0 elrond -> acme                portunreach
10/23/03    11:25:05    0 ES-s3.dante.bt.net -> acme    hostunreach
10/23/03    11:25:05    0 elrond -> acme                portunreach
10/23/03    11:25:39    0 elrond -> acme                portunreach
10/23/03    11:26:25    0 163.117.138.60 -> acme        portunreach
10/23/03    11:26:26    0 pc-11-58 -> acme              portunreach
10/23/03    11:26:45    0 arpa-gw.hpc.org -> acme       hostunreach
10/23/03    11:27:17    0 ES-s3.dante.bt.net -> acme    hostunreach
10/23/03    11:27:18    0 192.157.65.82 -> acme         hostunreach
10/23/03    11:27:41    0 elrond -> acme                portunreach
10/23/03    11:28:16    0 elrond -> acme                portunreach
10/23/03    11:28:16    0 arpa-gw.hpc.org -> acme       hostunreach
10/23/03    11:28:22    0 elrond -> acme                portunreach
10/23/03    11:28:26    0 192.157.65.82 -> acme         hostunreach
10/23/03    11:28:51    0 192.157.65.82 -> acme         hostunreach
10/23/03    11:29:05    0 192.157.65.82 -> acme         hostunreach
10/23/03    11:29:46    0 arpa-gw.hpc.org -> acme       hostunreach
```

Figura 3.4 Ejemplo del archivo de trazas generado por Icmplogger

3.2.4 Etherscan

Es una herramienta que monitorea la red buscando ciertos protocolos con actividad inusual, como puedan ser conexiones TFTP, comandos en el puerto de sendmail (25 TCP) como vrfy, expn, algunos comandos de rpc como rpcinfo, peticiones al servidor de NIS (algunas herramientas utilizan este tipo de servidores para obtener el archivo de password, ej: ypx), peticiones al demonio de mountd, etc.

Etherscan se ejecuta en modo promiscuo en la máquina utilizando (al igual que las anteriores) el NIT (Network Interface Tap de SunOs 4.1.x), y también el "Packet Filtering Interface" para realizar esas capturas.

Veamos al igual que en los casos anteriores un ejemplo de archivo de trazas:

```
04/25/03 14:32:29 [rpc] pc_12B14B.uc3m.es.1500 acme RPC lookup for: pcnfsd
04/25/03 14:32:29 [rpc] pc_12B14B.uc3m.es.1501 acme RPC lookup for: pcnfsd
04/25/03 16:05:57 [rpc] tony.1500 acme RPC lookup for: ypserv
04/25/03 16:06:01 [rpc] tony.1502 acme RPC lookup for: ypserv
05/25/03 19:51:01 [smtp]arapaima.uc3m.es.1033 acme vrfy jose@acme
05/09/03 17:53:34 [rpc] paco.1501 acme RPC lookup for: pcnfsd
05/10/03 10:38:16 [smtp]saruman.1339 selene.uc3m.es unknown cmd: hello
05/11/03 16:26:00 [rpc] balleste.1500 acme RPC lookup for: pcnfsd
05/11/03 17:30:26 [smtp] bruno.cs.colorado.edu.4671 elrond EXPN rivera
05/17/03 14:47:44 [smtp] elrond.2725 tidos.tid.es vrfy jason
05/19/03 09:22:17 [rpc]  paco.1501 acme RPC lookup for: ypserv
05/19/03 09:32:31 [smtp] tornasol.2748 acme vrfy jose
05/19/03 09:32:53 [smtp] tornasol.2748 acme vrfy jose@di
05/19/03 09:33:10 [smtp] tornasol.2748 acme vrfy jose@kk
05/26/03 09:29:13 [rpc]  pc_12B15.uc3m.es.1500 acme RPC lookup for:
pcnfsd
05/26/03 09:29:13 [rpc]  pc_12B15.uc3m.es.1501 acme RPC lookup for:
pcnfsd
09/26/03 09:32:23 [smtp] arapaima.uc3m.es.1063 elrond vrfy
postmaster@uc3m.es
09/26/03 10:02:00 [rpc]  paco.1500 acme RPC lookup for: ypserv
09/26/03 10:02:00 [rpc]  paco.1500 acme RPC lookup for: ypserv
```

Figura 3.5 Ejemplo del archivo de trazas generado por Etherscan

3.2.5 Nstat

Esta herramienta que originariamente fue diseñada para obtener estadísticas de uso de varios protocolos, se puede utilizar para detectar cambios en los patrones de uso de la red, que nos puedan hacer sospechar que algo raro está pasando en la misma.

Esta herramienta viene acompañada por dos utilidades que nos permiten analizar la salida que origina nstat, a saber: nsum, nload. La primera de ellas, nos da información de ciertos periodos de tiempo. La segunda, es un programa awk que produce una salida que puede ser vista de forma gráfica por herramientas como xvgr.

Para concluir este apartado, podemos decir que esta herramienta es muy útil para detectar ciertos tipos de ataques, tal como hemos reflejado anteriormente, así como dar una idea de qué tipo de protocolos están viajando por la red.

Además, tiene la ventaja de que al estar en modo promiscuo, con sólo tenerlo en una máquina del segmento se puede tener monitoreado todo el segmento en el que esté conectado.

Los archivos que generan el Tcplogger y el Udplogger pueden ser útiles también para detectar ataques de tipo SATAN o ISS, ya que en los archivos de trazas se aprecian intentos de conexión muy cortos en el tiempo a puertos (tcp o udp) de forma consecutiva.

Estos programas junto con el Icmplogger pueden guardar su información en ASCII o en formato binario. En este segundo caso, el programa dispone de una herramienta (extract) que permite consultar los archivos de trazas dándole patrones de búsqueda, como puede ser el tráfico desde una red concreta, los intentos de conexión a puertos específicos, etc.

Aarhus, es una herramienta de dominio público que permite auditar el tráfico IP que se produce en nuestra red, mostrándonos todas las conexiones del tipo indicado que descubre. Este programa se ejecuta

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

A la hora de leer esa información disponemos de una herramienta que incluye el software (llamado ra) y que nos permite también realizar filtros de visualización. Una característica de esta herramienta es la posibilidad de filtrar paquetes de acuerdo a las listas de acceso de los routers CISCO.

Es posible por tanto decirle que nos capture aquellos paquetes que no cumplen las reglas de la lista de acceso definida para esa interfaz del router. Como en el caso anterior (netlog) es posible ejecutar el comando en modo promiscuo (si lo que queremos es auditar todo nuestro segmento). Este programa divide las transacciones en cuatro grupos: TCP, UDP/DNS, MBONE, ICMP.

Los datos de la intervención de la transacción de la red que Argus genera han sido utilizados para una amplia gama de tareas incluyendo la gerencia de la seguridad, red Facturación y contabilidad, operaciones gerencia de la red y funcionamiento Análisis.

Algunos ejemplos de captura pueden ser:

argus -w NombreArchivoTraza &

argus -w ArchivoSalida ip and not icmp &

En el ejemplo a) le indicamos que nos capture todas las transacciones que se producen en nuestra subred y que lo almacene en un archivo. En el ejemplo b) todo el tráfico ip pero no el icmp.

Algunos ejemplos de utilización:

a) **ra -r ArchivoSalida tcp and host galileo**

b) **ra -C lista_acceso dst net 163.117.1.0**

En el ejemplo a) vemos todo el tráfico tcp (tanto de entrada como salida) en la máquina galileo.

En el ejemplo b) vemos en tiempo real todas las transacciones a la red 163.117.1.0 que violan la lista de acceso de ese interfaz del router.

A continuación, un pequeño ejemplo del archivo generado por esta utilidad:

```
Mon 10/23 11:25:36     ip      router4         <-      255.255.255.255  CON
Mon 10/23 11:31:58     ip      router4         <-      255.255.255.255  CON
Mon 10/23 11:24:02     udp     acme.4075       ->      acme.domai TIM
Mon 10/23 11:24:02     udp     acme.4076       ->      acme.domai TIM
Mon 10/23 11:24:02     udp     acme.4079       ->      acme.domai TIM
Mon 10/23 11:24:48     udp     acme.4081       ->      acme.domai TIM
Mon 10/23 11:24:49     udp     acme.4082       ->      acme.domai TIM
Mon 10/23 11:25:36     udp     julieta.2137    ->      acme.domai TIM
Mon 10/23 11:25:36     udp     julieta.2138    ->      acme.domai TIM
Mon 10/23 11:25:37     udp     karenina.2135   ->      acme.domai TIM
Mon 10/23 11:25:37     udp     karenina.2136   ->      acme.domai TIM
Mon 10/23 11:25:37     udp     karenina.2137   ->      acme.domai TIM
Mon 10/23 11:26:04     udp     a09-unix.1359   ->      acme.domai TIM
```

Figura 3.6 Ejemplo del archivo de trazas generado por Argus

3.2.6 IP Angel

Lucid Security's ipAngel pone la diferencia entre las tradicionales herramientas para detección de intrusos. Esta herramienta detecta y protege aplicaciones vulnerables sin duplicar la capacidad del firewall. Ofrece una completa protección a las redes.

3.2.6.1 Características

- Opera exclusivamente con Firewall – 1
- No duplica la capacidad de operar de Firewall – 1
- Ignora el tráfica que no esta atacando la vulnerabilidad de la red
- Se actualiza automáticamente en cuanto a los nuevos ataques
- Monitorea vulnerabilidades
- Protege de los ataques en tiempo real

3.2.6.2 Ventajas

- Ofrece una protección mejorada al actualizarse automáticamente y trabajar en tiempo real.

- Es económico en relación a las otras herramientas que se encuentran actualmente en el mercado.
- Reduce tiempo al no hacer que se duplique la capacidad del Firewall.
- Es de fácil manejo y despliegue.
- Detecta y educa al firewall para que suprima y deshabilite los puntos vulnerables.

3.3 ISS (Internet Security Scanner)

Es una herramienta de la cual existe versión de dominio público que chequea una serie de servicios para comprobar el nivel de seguridad que tiene esa máquina.

ISS es capaz de chequear una dirección IP o un rango de direcciones IP (en este caso se indican dos direcciones IP e ISS chequeará todas las máquinas dentro de ese rango).

El programa viene acompañado de dos utilidades que son ypx y strobe. La primera, nos permite la transferencia de mapas NIS a través de la red y la segunda, chequea y describe todos los puertos TCP que tiene la máquina que chequeamos.

Con la primera herramienta es posible la transferencia de los archivos de "password" en aquellas máquinas que hayan sido configuradas como servidores de NIS. ISS se puede ejecutar con varias opciones y la salida se

deja en un archivo. Además, si ha podido traerse el archivo de "password" de la máquina chequeada, creará un archivo aparte con la dirección IP de la máquina.

3.3.1 ¿Cuál es la diferencia entre él e ISS y ANGEL?

El ISS, y cualquier otra herramienta de revisión realizan revisiones a una red y advierte sobre cualquier problema que puede encontrar.

Mientras que ANGEL actúa al chequear automáticamente los blancos secundarios.

3.4 Módulos de Autentificación Conectables (PAM)

Este programa ofrece privilegios a los usuarios deben autentificarse adecuadamente.

Al iniciar una sesión en un sistema, el usuario proporciona su nombre de usuario y contraseña y el procedimiento de inicio de sesión usa el nombre de usuario y la contraseña para autentificar el inicio de sesión para verificar que el usuario es quien dice ser.

Son posibles otras formas de autentificación además de las contraseñas. Los Pluggable Authentication Modules (PAM) es una manera de permitir que el administrador de sistema establezca una política de autentificación sin tener que recompilar programas de autentificación.

3.4.1 Ventajas de PAM

PAM provee muchas ventajas para un administrador de sistema, como las siguientes:

- Un esquema de autentificación común que se puede usar con una gran variedad de aplicaciones.
- PAM puede ser ejecutado con varias aplicaciones sin tener que recompilar las aplicaciones para soportar PAM específicamente.
- PAM permita una gran flexibilidad y control cuando de autentificación para el administrador y para el desarrollador de aplicaciones se refiere.
- Los desarrolladores de aplicaciones no tienen la necesidad de desarrollar su programa para usar un determinado esquema de autentificación. En su lugar, pueden concentrarse puramente en los detalles de su programa.

3.5 Kerberos

Es un servicio de autenticación basado en el protocolo de distribución de claves. El objetivo principal que tiene Kerberos es, proporcionar un sistema que permita la autenticación en la comunicación entre clientes y servidores, y de esta forma evitar que los passwords de los usuarios viajen continuamente por la red. El sistema se basa en una serie de intercambios cifrados, denominados "tickets" o vales, que permiten controlar el acceso desde las estaciones de trabajo a los servidores.

Proporciona, asimismo, una serie de verificaciones criptográficas para garantizar que los datos transferidos entre estaciones y servidores no estén corrompidos, bien por accidente o bien por ataques intencionados.

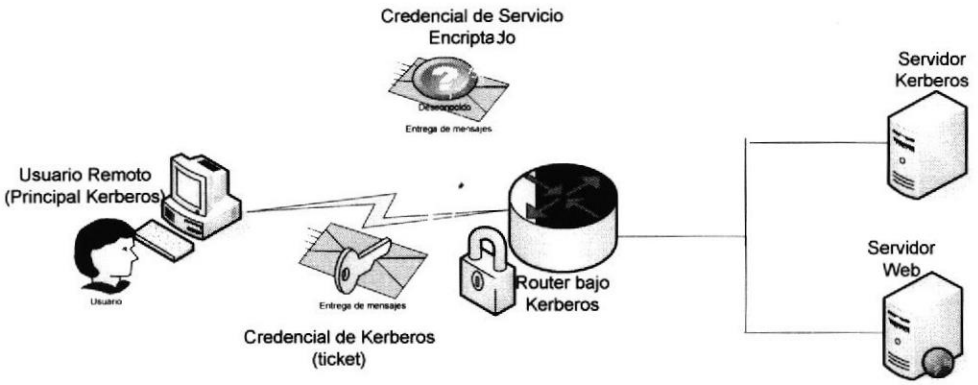


Figura 3.7 Funcionamiento de Kerberos

3.6 SSH (o Secure Shell)

SSH es un protocolo para crear conexiones seguras entre dos sistemas.

Usando SSH, la máquina del cliente inicia una conexión con una máquina de servidor. SSH proporciona los siguientes tipos de protección:

- Después de la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor durante sesiones ulteriores.
- El cliente puede transmitir su información de autentificación al servidor, como el nombre de usuario y la contraseña, en formato cifrado.

- Todos los datos enviados y recibidos durante la conexión se transfieren por medio de encriptación fuerte, lo cual los hacen extremamente difícil de descifrar y leer.

El cliente tiene la posibilidad de usar X11 aplicaciones lanzadas desde el indicador de comandos de la shell. Esta técnica proporciona una interfaz gráfica segura (llamada reenvío por X11).

El servidor también obtiene beneficios por parte de SSH, especialmente si desempeña una cierta cantidad de servicios. Si usa el reenvío por puerto, los protocolos que en otros casos serían considerados inseguros (POP, por ejemplo) se pueden cifrar para garantizar comunicación segura con máquinas remotas. SSH hace relativamente sencilla la tarea de cifrar tipos diferentes de comunicación que normalmente se envía en modo inseguro a través de redes públicas.

3.7 Nessus

Es la herramienta de evaluación de seguridad un escáner de seguridad remoto para Linux. Está basado en plug-in(s), tiene una interfaz basada en GTK, y realiza más de 1200 pruebas de seguridad remotas.

Permite generar reportes en HTML, XML, LaTeX, y texto ASCII; también sugiere soluciones para los problemas de seguridad. Nessus revisará remotamente una red dada y se determinará si alguien (o algo - como un gusano) puede romperse en él, o lo emplea mal de una cierta manera.

Nessus tiene la capacidad de probar los servicios de SSLized tales como https, smtps, imaps, y más.

3.8 Snort

Un sistema de detección de intrusiones (IDS) libre para las masas. Snort es un sistema de detección de intrusiones de red de poco peso (para el sistema), capaz de realizar análisis de tráfico en tiempo real y registro de paquetes en redes con IP.

Puede realizar análisis de protocolos, búsqueda / identificación de contenido y puede ser utilizado para detectar una gran variedad de ataques y pruebas, como por Ej. buffer overflows, escaneos indetectables de puertos, ataques a CGI, pruebas de SMB, intentos de reconocimientos de sistema operativos y mucho más.

Snort utiliza un lenguaje flexible basado en reglas para describir el tráfico que debería recolectar o dejar pasar, y un motor de detección modular. Mucha gente también sugirió que la Consola de Análisis para Bases de Datos de Intrusiones (Analysis Console for Intrusion Databases, ACID) sea utilizada con Snort.

3.9 Ethereal

Con este analizador de protocolos de red podemos examinar datos de una red viva o de un archivo de captura en algún disco. Se puede examinar

interactivamente la información capturada, viendo información de detalles y sumarios por cada paquete.

Ethereal tiene varias características poderosas, incluyendo un completo lenguaje para filtrar lo que querramos ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP. Incluye una versión basada en texto llamada tethereal.

Utilizado para localizar averías, desarrollo del análisis, del software y del protocolo, y educación. Hace todas las características de estándar que se espera en un analizador del protocolo.

3.10 Netfilter

Es un poderoso filtro de paquetes el cual es implementado en el kernel Linux estándar.

La herramienta iptables es utilizada para la configuración. Actualmente soporta filtrado de paquetes stateless o statefull, y todos los diferentes tipos de NAT (Network Address Translation) y modificación de paquetes.

Entre las características de esta herramienta tenemos:

- filtración apátrida del paquete (IPv4 e IPv6)
- filtración stateful del paquete (IPv4)
- todas las clases de la dirección de red y de la traducción del puerto (NAT/NAPT)

- infraestructura flexible y extensible
- capas múltiples de los APÍs para las extensiones de la tercer persona
- el número grande de plugins/modules mantuvo depósito del ' remiendo-o-matic '

3.11 GnuPG / PGP

PGP programa de encripción diseñado que ayuda a proteger nuestra información de curiosos y otros riesgos. De esta manera protegemos nuestros archivos y comunicaciones con cifrado avanzado.GnuPG.

Motor de cifrado, en sí mismo, que puede ser utilizado directamente desde la línea de comandos, desde programas de shell (shell scripts) o por otros programas. Por lo tanto GnuPG puede ser considerado como un motor (backend) para otras aplicaciones.

El conjunto de órdenes de esta herramienta siempre será un super conjunto del que proporcione cualquier interfaz de usuario como:

- Reemplazo completo de PGP.
- No utiliza algoritmos patentados.
- Con licencia GPL, escrito desde cero.
- Puede utilizarse como filtro.
- Facilidad de implementación de nuevos algoritmos utilizando módulos.

- El identificador de usuario (User ID) es obligado a estar en un formato estándar.
- Soporte integrado para servidores de claves HKP (wwwkeys.pgp.net).

3.12 Stunnel

Stunnel es una envoltura criptográfica SSL de propósito general. Diseñado para trabajar como una envoltura de cifrado SSL entre un cliente remoto y un servidor local (ejecutable por inetd) o remoto. Stunnel puede ser utilizado para agregarle funcionalidad SSL a daemons utilizados comúnmente como POP2, POP3, y servidores de IMAP sin cambios en el código del programa. Negocia una conexión SSL utilizando la biblioteca de OpenSSL o la SSLeay.

Stunnel permite cifrar conexiones arbitrarias del TCP dentro del SSL. Stunnel permite asegurar a protocolos (IMAP, LDAP, etc) proporciona el cifrado, sin realizar cambio al código del demonio.

Requiere una biblioteca de funcionamiento del SSL tal como OpenSSL para compilar el stunnel.

Stunnel negociará conexiones del SSL entre el cliente y el servidor. Puesto que todo el código crypto se contiene en las bibliotecas del SSL, su compilación de las bibliotecas dichas dictará qué algoritmos serán utilizados.

3.13 Filtro Anti-Spam

Crear un Filtro Anti-Spam creando listas negras con las cuentas de correo desde donde te llega el spam. De esta manera se evitará que lleguen dichos mensajes al gestor de correo de los usuarios, y eliminarlos desde el servidor.

Por lo general esta herramienta Anti-Spam se usa con el software WebMail, pero está comprobado que funciona con otros clientes de correo convencional (Netscape messenger, Outlook Express, Eudora, etc...).

En este caso, deberá de conocer su contraseña de correo, que es distinta de la de UPVNet. Si la desconoce, cámbiela desde sus páginas personalizadas. Configure su cliente de correo de acuerdo a las consideraciones siguientes:

3.13.1 Correo entrante (POP o IMAP)

Como protocolo puede utilizar POP o IMAP.

- Con POP los mensajes que se reciben en la carpeta de entrada son retirados del servidor y se transfieren a directorios locales de su ordenador.
- Con IMAP los mensajes quedan en las carpetas del servidor y pueden ser consultados también desde otros ordenadores o programas de correo, incluido el webmail.

Es necesario configurar el uso de conexiones seguras (TSL o SSL), lo que implica el uso de protocolos criptados. Esta opción puede hacer algo lenta las consultas pero garantiza la confidencialidad de sus datos.

3.13.2 Servidor de correo saliente (SMTP)

Como servidor de correo saliente debe de indicar el que le suministre su proveedor de acceso a internet (ISP).

3.14 Motor de búsqueda (Buscador)

Esta característica permite instalar y administrar la Máquina de Búsqueda WebGlimpse desde el sitio, de tal forma que permitirá a los usuarios llevar a cabo búsquedas en el sitio.

Webglimpse es un motor de búsqueda que se ha utilizado en los millares de sitios. Casi todo sobre Webglimpse es configurable: cómo seleccionar los archivos para poner en un índice, cómo buscarlos, y cómo presentar los resultados al usuario.

Todavía, hemos intentado mantener la instalación simple y rápida de modo que usted pueda conseguir su búsqueda en servicio hoy.

Construye un índice de la palabra clave por adelantado para buscar muy rápido (aunque puede también tener acceso a los archivos individuales para las preguntas booleanas complejas).

Las palabras infrecuentes serán encontradas rápidamente incluso en un fileset muy grande, hasta varios gigabytes. Las palabras comunes (con 100 o 1000's de fósforos) durarán, pero si el número de los golpes vueltos puede ser limitado, incluso ésos estarán muy rápidamente.

Utiliza un índice pequeño - típicamente menos el de 5% del tamaño total de los datos - así que puede ser cargada generalmente enteramente en memoria. Probablemente el punto más fuerte de Webglimpse - extremadamente flexible, las reglas configurables para el cual archivan para poner en un índice, cómo buscar, concordancia con el modelo, graduación del resultado.

Webglimpse no se limita a buscar solamente sus propios datos. El programa de la araña tiene reglas flexibles para recolectar las páginas de sitios alejados. Puede recolectar todas las sin importar páginas bajo dominio especificado, o travieso un número del sistema de los saltos de una página que comienza dominio, o una combinación de estas reglas. Usted puede incluso hacer sobre duro solo archivo, investigable a partir de una forma, que combina datos locales un su impulsión y los sitios alejados múltiples que usted especifica.

3.15 FTP anónimo

FTP anónimo permite compartir ficheros con los usuarios sin necesidad de darles claves para la conexión. Dispone de una carpeta donde los usuarios pueden dejar ficheros también.

Existen varias implantaciones del protocolo de FTP disponibles. Implementaremos el FTP con un programa que se llama WS_FTP Limited Edition (o WS_FTP Pro, el cual no es gratis, es la versión requerida para uso comercial).

- Entrando en el Servidor
- Buscando el Archivo
- Bajando un Archivo

3.16 Servidor SSL

Es recomendable utilizar el servidor web Apache-SSL en todos los servidores. Con el se puede colocar en cualquier área del web directorios https seguros para proteger áreas que contienen información sensible.

La mayoría de las "teóricas" transacciones seguras de comercio electrónico en Internet se realizan mediante SSL. Está basado en el clásico cifrado de clave pública que puede parecer "intimidatorio" para un hacker novato, pero se puede entender de la siguiente manera: cuando el cliente pide al servidor seguro una comunicación segura, el servidor abre un puerto cifrado, gestionado por un software llamado Protocolo SSL Record, situado encima de TCP.

Será el software de alto nivel, Protocolo SSL Handshake, quien utilice el Protocolo SSL Record y el puerto abierto para comunicarse de forma segura con el cliente.

3.16.1 El Protocolo SSL Handshake

Durante el protocolo SSL Handshake, el cliente y el servidor intercambian una serie de mensajes para negociar las mejoras de seguridad.

Este protocolo sigue las siguientes seis fases:

- La fase Hola, usada para ponerse de acuerdo sobre el conjunto de algoritmos para mantener la intimidad y para la autenticación.
- La fase de intercambio de claves, en la que intercambia información sobre las claves, de modo que al final ambas partes comparten una clave maestra.
- La fase de producción de clave de sesión, que será la usada para cifrar los datos intercambiados.
- La fase de verificación del servidor, presente sólo cuando se usa un algoritmo de intercambio de claves, y sirve para que el cliente autentique al servidor.
- La fase de autenticación del cliente, en la que el servidor solicita al cliente un certificado X.509 (si es necesaria la autenticación de cliente).
- Por último, la fase de fin, que indica que ya se puede comenzar la sesión segura.

3.16.2 El Protocolo SSL Record

El Protocolo SSL Record especifica la forma de encapsular los datos transmitidos y recibidos. La porción de datos del protocolo tiene tres componentes:

- MAC-DATA, el código de autenticación del mensaje.
- ACTUAL-DATA, los datos de aplicación a transmitir.
- PADDING-DATA, los datos requeridos para rellenar el mensaje cuando se usa cifrado en bloque.

Actualmente el protocolo de cifrado más potente disponible es de 128 bits y está implementado en todas las versiones de los actuales navegadores.

Hasta hace unos años esto no era posible debido a que las leyes de exportación de los EEUU consideraban a las claves de cifrado mayores de 40 bits, como "material de guerra".

Respecto a la violación de un certificado SSL de un sitio web legítimo. En teoría, esta violación sería invalidada gracias a una verificación cruzada de la identidad del certificado con el nombre DNS y dirección IP del servidor que se encuentra al otro extremo de la conexión. Esta es la teoría de la especificación SSL.

Engañando al explorador para que abra una sesión SSL en un servidor web malintencionado que se haga pasar por otro servidor legítimo, todas las

sesiones SSL posteriores que se establezcan con el servidor web legítimo terminarían finalmente en el servidor malintencionado sin que el usuario recibiera ninguno de los mensajes de aviso normales.

3.17 SpamGuard

Incluir esta herramienta para evitar que lleguen correos electrónicos no deseados. ¿Cómo?

- Se puede bloquear aquellas cuentas de correo desde donde llegan mensajes que no se quieren.
- Se puede crear una lista de palabras con las que censurar correos. Es decir, determinar que si llegan mensajes donde aparece una palabra o frase en concreto, directamente se elimine o se almacene en un fichero especial.
- De esta forma El Spam no afectará a la trasferencia y el espacio de buzones.

3.18 Software de Proxy

Es necesario determinar un software de Proxy algunos son a nivel de aplicación (como SQUID) y otros son a nivel de sesión (como SOCKS). Squid es un proxy a nivel de aplicación para HTTP, HTTPS y FTP. También puede ejecutar peticiones DNS bastante más rápido de lo que puede hacerlo la mayoría del software cliente. SQUID es ideal para acelerar el acceso a la internet, y para controlar el acceso a sitios web (utilizando

paquetes como squidGuard). Entre otras cosas, Squid puede hacer Proxy y cache con los protocolos HTTP, FTP, GOPHER y WAIS, Proxy de SSL, cache transparente, WWCP, aceleración HTTP, cache de consultas DNS y más.

Squid apoya:

- Proxying y el depositar del HTTP, del ftp, y del otro URL
- El proxying para el SSL
- Jerarquías del escondrijo
- IPC, HTCP, CARPA, Resúmenes Del Escondrijo
- Controles de acceso extensos
- Aceleración del servidor del HTTP
- SNMP
- El depositar de las operaciones de búsqueda del DNS

3.19 Metodología para la detección de intrusos

La labor de un administrador o de la persona encargada de la seguridad puede ser realmente frustrante. Sobre todo cuando el sistema ha sido invadido por un intruso o hacker.

En principio, si se ha configurado correctamente un servidor y se está al día en materia de seguridad, así como de fallas que van surgiendo, no habrá problemas de que un intruso entre en el sistema. Realmente con un poco de esfuerzo se puede tener un servidor altamente seguro que evitará alrededor del 85% de los intentos de acceso no autorizados alo sistema.

Pero en muchas ocasiones el peligro viene de los propios usuarios internos del sistema, los cuales presentan un gran riesgo debido a que ya tiene acceso al sistema.

3.19.1 Pasos a seguir para detectar a un intruso

Lo primero que debemos hacer es seguir una serie de pasos los cuales nos ayudarán a descubrir si realmente ha entrado un intruso, ya que en muchas ocasiones pensamos que ha entrado alguien, pero no es cierto. Por eso, ante todo calma; esto es lo más importante para un buen administrador.

Este esquema representa básicamente los pasos que sigue de un intruso: Primero entra al sistema, y si sólo tiene acceso como usuario, explotará alguna debilidad o falla del sistema para así obtener ID 0 (o lo que es lo mismo, privilegios de root).

En caso de entrar como root u obtenerlo de alguna otra manera, se dedicará a controlar el sistema, dejando algún mecanismo para volver cuando quiera.

Seguramente copiará el archivo /etc/passwd y el /etc/shadow (en caso de que el sistema use "shadow"), luego le dará rienda suelta a su imaginación, como por ejemplo, instalar un sniffer, troyanos, leer mails ajenos, etc. Y en caso de ser un pirata malicioso puede causar desastres en el sistema, como sería modificar paginas web, borrar archivos o

mails, producir un DoS (Denial of Service), cambiar passwords de usuarios legítimos, etc.

Figura 3.8 Pasos que sigue un intruso

1. Examinar los archivos log como el 'last' log, contabilidad, syslog, y los C2 log buscando conexiones no usuales o cosas sospechosas en el sistema. Aunque hay que tener especial cuidado en guiarnos por los logs, ya que muchos intrusos utilizaran diversas herramientas para borrar sus huellas.
2. Buscar por el sistema archivos ocultos o no usuales, ya que pueden ser usados para esconder herramientas para violar la seguridad del sistema, por ejemplo un crackeador o incluso contener el /etc/passwd del sistema o de otros sistemas al cual ha entrado nuestro intruso.
3. Buscar archivos SET-UID por el sistema. Ya que en muchas ocasiones los piratas suelen copiar y dejar escondido copias del /bin/sh para obtener root. Podemos utilizar el comando 'find' para buscar este tipo de archivos por el sistema (el comando 'find' puede ser sustituido por un troyano para esconder archivos del pirata, por lo que no es totalmente confiable), para ello ejecutamos la siguiente línea: # find / -user root -perm -4000 -print

4. Revisar los archivos binarios del sistema para comprobar que no han sido sustituidos por un troyano, como por ejemplo los programas 'su', 'login', 'telnet' y otros programas vitales del sistema. Lo recomendado es comparar con las copias de seguridad aunque puede que las copias de seguridad también hayan sido sustituidas por un troyano.
5. Examinar todos los archivos que son ejecutados por 'cron' y 'at'. Ya que algunos piratas depositan puertas traseras que le permiten volver al sistema aunque los hayamos echado del sistema. Asegurarse que todos los archivos son nuestros y no tienen permiso de escritura.
6. Examinar el archivo /etc/inetd.conf en busca de cambios, en especial aquellas entradas que ejecuten un shell y comprobar que todos los programas son legítimos del sistema y no troyanos.
7. Examinar los archivos del sistema y de configuración en busca de alteraciones. En particular, buscar entradas con el signo '+' o 'host names' no apropiados en archivos como /etc/hosts.equiv, /etc/hosts.lpd y en todos los archivos .rhost del sistema, con especial interés los de 'root', 'uucp', 'ftp' y otras cuentas del sistema. Estos archivos no deberían tener atributo de escritura.
8. Examinar cuidadosamente todos los computadores de nuestra red local en busca de indicios que nuestra red ha sido comprometida. En particular, aquellos sistemas que compartan NIS+ o NFS, o aquellos sistemas listados en el /etc/hosts.equiv. Lógicamente también

revisar los sistemas informáticos que los usuarios comparten mediante el acceso del .rhost.

9. Examinar el archivo /etc/passwd, en busca de alteraciones en las cuentas de los usuarios o la creación de cuentas nuevas, especialmente aquellas cuentas con ID 0, las que no tienen password, etc.

Estos nueve puntos son los pasos a seguir recomendado, los cuales están muy bien, pero se quedan un poco cortos de soluciones practicas para el administrador. Para ello nos vamos a basar en un excelente trabajo de un grupo de Hackers conocidos como Technotronic, los cuales se basan en el mismo documento pero explicando los métodos de los piratas y como combatirlos. Se trata de lo siguiente:

Los archivos Log:

- **messages:** Este archivo contiene bastante información, por lo que debemos buscar sucesos poco usuales.
- **xferlog:** Si el sistema comprometido tiene servicio FTP, este archivo contiene la bitácora de todos los procesos del FTP. Podemos examinar qué tipo de herramientas ha subido el pirata y qué archivos ha bajado de nuestro servidor.
- **utmp:** Este archivo contiene información en binario de todos los usuarios conectados al sistema en el momento. Por lo que puede

ser muy útil para determinar quién esta conectado al sistema en este momento. Para ello ejecutaremos el comando 'who' o 'w'.

- **wtmp:** Cada vez que un usuario entra al servidor y sale del mismo, la máquina modifica este archivo. Al igual que el anterior, este archivo está en binario, por lo que tendremos que usar alguna herramienta especial para ver el contenido de este archivo. El mismo contiene la información en formato: usuario, hora de conexión, e IP origen del usuario, por lo que podemos averiguar de dónde provino el pirata. Pero aunque contemos con esta información, puede que haya sido falseada por el pirata utilizando alguna técnica para ocultar su IP original o haya borrado su entrada.

Muchos piratas intentaran borrar sus huellas utilizando unos programas conocidos como 'Zapper's' o 'Zap'. Los más populares, debido a que están ampliamente disponibles por Internet, son los siguientes:

- marry.c
- zap.c
- zap2.c
- remove.c
- cloak.c

Este ultimo comando 'ncheck' nos permitirá buscar archivos SETUID por las particiones. Debemos buscar troyanos en nuestros archivos binarios,

ya que suele ser una de las tareas principales de un pirata cuando ha comprometido la seguridad de un servidor. Una lista no exhaustiva de posibles binarios que un pirata puede sustituir, es la siguiente:

- login
- su
- telnet
- netstat
- ifconfig
- ls
- find
- du
- df
- libc
- sync

Así como los binarios listados en /etc/inetd.conf.

Al igual que antes, tenemos varias utilidades ampliamente disponibles para detectar estos troyanos, pero por otro lado, los piratas también tienen ampliamente disponible estos paquetes de troyanos, conocidos como 'RootKit', como ya mencionamos antes. Otras de las principales

tareas de un pirata consisten en la utilización de sniffers, para capturar información confidencial. Los más usados son los siguientes:

- linsniff666.c
- esniff.c
- solsniff.c
- sunsniff.c
- sniffit

Un pirata intentará por todos los medios obtener el archivo de passwords, para luego usar un programa especial que le permitirá averiguar los passwords de los usuarios. Los principales 'crackeadores' son:

- Crack
- John The Ripper 1.5
- Cracker Jack
- Hades

3.19.2 Pasos a seguir cuando hemos detectado un pirata

Si hemos pillado al intruso en el momento, tenemos varias opciones:

- Hablar con él, usando el comando 'talk', aunque debemos tener en cuenta que puede contestar de forma amistosa (ayudándonos en

relación a la seguridad del sistema) o agresiva (borrando el sistema para no dejar rastro).

- Desconectarle del sistema, usando el comando 'kill', pero para evitarnos que vuelva a entrar, antes de usar 'kill', usaremos el comando 'passwd' para cambiar el password de la cuenta por el cual el pirata entró.
- Utilizar las utilidades del sistema para recopilar información sobre el pirata, la cual será necesaria en caso de denuncia. Por lo que trataremos de 'tracearle', usando los siguientes comandos:
  - who
  - w
  - last
  - lastcomm
  - netstat
  - snmpnetstat
  - Obtener información del router.
  - Examinar el archivo /var/adm/messages.
  - Examinar el syslog.
  - Examinar los log del wrapper.
  - Ejecutar el comando 'finger' en todos los usuarios locales, para comprobar cuando fue la ultima vez que estuvieron en el sistema.
  - Examinar los archivos history del shell, como el .history, .rchist y archivos similares.

- Ejecutando el comando 'finger', intentaremos sacar información del host de donde provino el ataque, como por ejemplo:

Si tenemos suerte podremos obtener la información.

- También podemos dirigirnos a Internic (http://www. internic.net) donde podemos pedir información sobre cualquier servidor en el mundo, siempre y cuando no sea militar. Allí ponemos el dominio del servidor de donde provino el ataque, y podremos ver con quién debemos ponernos en contacto con el servidor atacante.
- Ahora que tenemos bastante información del atacante, lo mejor sería desconectar nuestro servidor de Internet y dedicarnos unos días a repasar cuidadosamente lo sucedido. Además, si nadie accede al servidor durante unos días, podremos trabajar mejor y más rápido.
- Tendremos que hacer una copia de seguridad. En caso de duda de cómo usar el comando 'dd', lo mejor sería recurrir al comando 'man'.
- Tener a mano un block de notas y un lápiz, para escribir todo lo que nos parezca importante, así como todos los pasos que estamos realizando en el sistema comprometido.

Pasos a seguir para recuperar el control en nuestro sistema que hay que mirar con detalle.

1. Creación de cuentas nuevas o alteración de algunas existentes.
2. Excesivo consumo de memoria o disco duro.
3. Directorios o archivos sospechosos.
4. Alteración en la configuración del sistema.
5. Procesos sospechosos.
6. Conexiones de servidores poco usuales.
7. Reconfiguración de los modems.
8. Serie de repeticiones de conexión al mismo puerto.
9. Conexión de usuarios en horas o días poco usuales.

3.19.3 Hackers e intrusos en las organizaciones

De acuerdo a unas encuestas realizadas en los primeros meses del año 2005 por la empresa KPMG de Argentina se pudo constatar que no todos aquellos empresarios tienen la certeza de la seguridad de sus redes. Se realizó una consulta a un grupo de empresarios en la que se les consultó si sus redes habían sido atacadas por hackers y de acuerdo a sus respuestas nos indicaron lo siguiente:

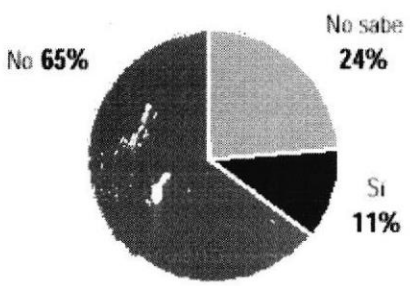


Figura 3.9 Porcentajes obtenidos de la Encuesta si han sido atacados por hackers

A aquellas personas que respondieron de manera afirmativa en cuanto a ataques se les preguntó sobre cuales habían sido los objetivos del mismo y los resultados de sus respuestas las vemos a continuación:

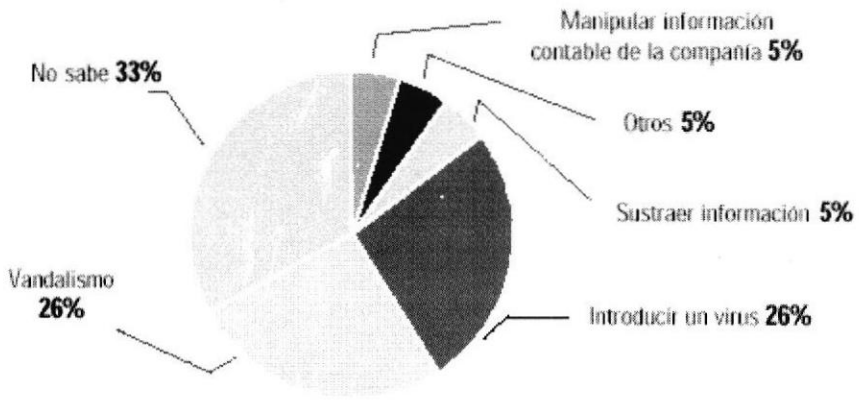


Figura 3.10 Porcentajes obtenidos de la Encuesta acerca de los objetivos del ataque

Y en esta encuesta se pudo verificar que no solo redes de empresas pueden ser atacadas por hackers, sino también personas que poseen computadoras personales, por lo que los resultados fueron los siguientes:

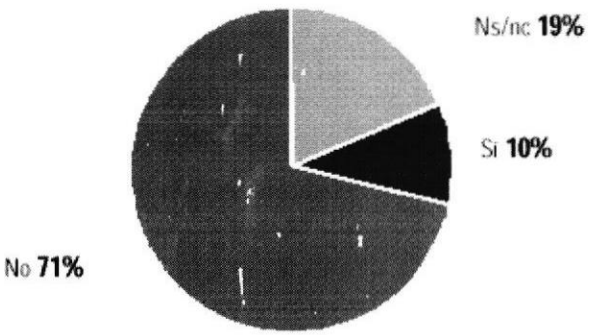


Figura 3.11 Porcentajes obtenidos de la Encuesta de ataques a computadoras personales

Al igual que a los empresarios, a aquellos que respondieron de manera afirmativa atribuyen los ataques fundamentalmente a los siguientes propósitos:

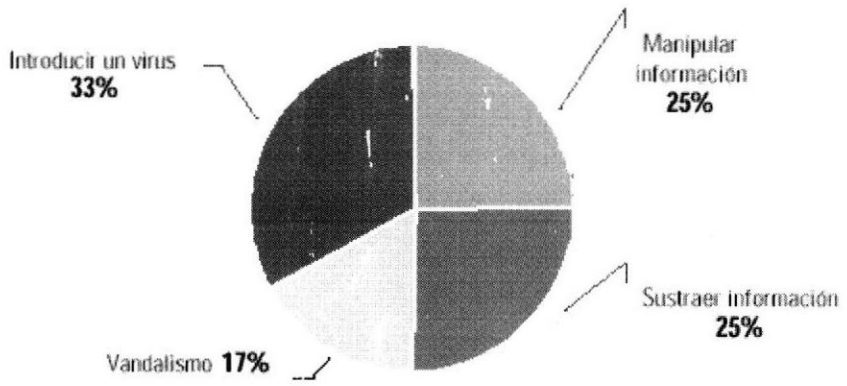


Figura 3.12 Porcentajes obtenidos acerca de los objetivos del ataque a computadoras personales

CAPÍTULO 4

4. TECNOLOGÍAS QUE PERMITEN GENERAR CONTENIDO DINÁMICO

4.1 ASP (Active Server Pages)

Esta herramienta permite generar contenido dinámico y acceder a base de datos de una manera muy sencilla, y sin la necesidad de tener conocimientos de programación. Puedes programar fácilmente con ASP usando Microsoft FrontPage, Visual InterDev, Macromedia Dreamweaver Ultradev y Adobe Golive. Incluye los siguientes objetos:

- **Objeto Application:** el objeto Application se utiliza para compartir información entre todos los usuarios de una misma aplicación.
- **Objeto Request:** el objeto Request se utiliza para tener acceso a la información que se pasa en las peticiones HTTP.

  Entre dicha información se incluyen los parámetros que se pasan desde los formularios HTML mediante el método POST o el método GET, cookies y certificados de cliente.
- **Objeto Response:** el objeto Response se utiliza para controlar la información que se envía al usuario.

  Esto incluye el envío de información directamente al explorador, la redirección del explorador a otra dirección URL o el establecimiento de valores de las cookies.

- **Objeto Server:** el objeto Server proporciona acceso a los métodos y las propiedades del servidor.

  El método utilizado con más frecuencia en Servidores Web es aquel que crea una instancia de un componente ActiveX (Server. CreateObject).

- **Objeto Session:** el objeto Session nos permite almacenar la información necesaria para una determinada sesión de usuario.

  Las variables almacenadas en el objeto Session no se descartan cuando el usuario pasa de una página a otra dentro de la misma aplicación, cuando esto sucede dichas variables persisten durante todo el tiempo que el usuario tiene acceso a las páginas de la aplicación.

También puede utilizar los métodos de Session para terminar explícitamente una sesión y establecer el periodo de tiempo de espera de inactividad de las sesiones.

4.2 JSP Java Server Page

JSP es una interfaz de programación de aplicaciones de servidores Web. En una página JSP se combinan bloques de HTML estáticos, y HTML dinámico generados con Java que se ejecutan en el servidor.

Una página JSP puede procesar formularios Web, acceder a bases de datos y redireccionar a otras páginas. Las páginas JSP son transformadas

a un Java Servlet y después compiladas. El contenedor JSP proporciona un motor que interpreta y procesa una página JSP como un servlet. Al estar basadas en los servlets, las distintas peticiones a una misma página JSP son atendidas por una única instancia del servlet.

4.2.1 Diferencias entre JSP y Servlet

En JSP, el código de presentación está separado de la lógica del programa, mientras que en un servlet, el código de presentación se compila dentro de la clase. En una página JSP el código de presentación puede ser actualizado por un diseñador web que no conozca Java. Los servlets se encuentran ya compilados, mientras que las páginas JSP se compilan bajo petición, lo que hace que la ejecución del servlet sea algo más rápida.

4.2.2 Características de JSP

Permiten separar la parte dinámica de la estática en una página web. Las páginas JSP se almacenan en el servidor en archivos con extensión .jsp. El código JSP es java y se encierra entre: <% y %>, por ejemplo:

```
<H1>Hora: <%= new java.util.Date() %></H1>
```

La síntaxis también se puede expresar en formato XML

```
<jsp:xxx> ... </jsp:xxx>
```

En una página JSP hay varios objetos implícitos: request, response, out, session, application, config, pageContext, page y exception.

Cada página JSP es compilada automáticamente hacia un servlet por el motor JSP la primera vez que se accede a esa página. Desde una página JSP se puede llamar a un componente JavaBean donde se puede implementar la lógica de negocio.

4.2.3 Funcionamiento de JSP

Una página JSP es básicamente una página Web con HTML tradicional y código Java incrustado. La extensión de fichero de una página JSP es ".jsp" en vez de ".html" o ".htm", y eso le indica al servidor que esta página requiere un tratamiento especial que se conseguirá con una extensión del servidor o un plug-in.

El servidor web comprueba si la página ha sido solicitada con anterioridad. En ese caso el servlet correspondiente ya estará cargado en memoria. Si no es así, se notifica al motor de jsp y se generará un servlet para la página.

Cuando un cliente solicita una página JSP, se ejecuta en el servidor el código JSP de la página, dando como resultado una página HTML que se fusiona con el HTML original, generando una página HTML de respuesta que será enviada al cliente.

4.3 CGI Common Gateway Interface

CGI son las siglas de Common Gateway Interface, o interfaz de pasarela común. Se trata de una especificación que va a realizar la función de interfaz o pasarela entre el servidor web y los programas, llamados programas CGI, haciendo uso del protocolo HTTP y el lenguaje HTML. Un programa CGI será aquel que cumpla la especificación CGI, es decir, interactuará con el servidor de acuerdo a unos principios establecidos en la especificación. Veamos cómo funciona esto.

Usualmente, cuando un navegador busca un URL, sucede lo siguiente. En primer lugar, el ordenador cliente contacta con el servidor HTTP. Este busca el fichero solicitado por el cliente y envía ese fichero. El cliente entonces visualiza el fichero en el formato apropiado. Ahora bien, es posible instalar el servidor HTTP de forma que cuando un fichero de un directorio concreto es solicitado, ese fichero no sea devuelto. En lugar de eso, se ejecuta como un programa, y todo lo que el programa obtiene se envía de vuelta al cliente para ser visualizado. Obviamente, el directorio en el que están estos programas debe tener permiso de ejecución, así como los programas, y los permisos de lectura o de lectura/escritura para otros programas que pudieran usarse.

En resúmen los programas CGI son programas que se ejecutan en el servidor en respuesta a peticiones del cliente. El servidor creará una información especial para el CGI cuando pasa a ejecutarlo, y esperará la respuesta del programa. Antes de que el CGI se ejecute, el servidor crea

un entorno con el que trabajará el programa CGI. Este entorno comprende la traducción de cabeceras de peticiones HTTP en variables de entorno a las que podrá acceder nuestro programa.

El resultado de la ejecución del programa suele ser una serie de encabezados de respuesta HTTP y HTML. Estos encabezados son recogidos por el servidor y enviados al cliente que hizo la petición.

Los programas o scripts CGI pueden escribirse en cualquier lenguaje de programación que sepa manejar entrada y salida estándar. La elección depende de qué nos gusta más, y un poco de sobre qué sistema operativo está el servidor. Si el servidor corre bajo una máquina Unix, a buen seguro podremos programar en C o en Perl, solicitando al administrador del sistema que le de los permisos necesarios para poder ejecutar los programas.

Si el servidor corre bajo una máquina Windows, también podremos programar en C o en Perl, esto último si el servidor tiene el intérprete instalado. Lo que programemos en C tendremos que compilarlo y poner el ejecutable en el directorio destinado a los CGI.

Si usamos Perl o algún otro lenguaje interpretado, no tendremos necesidad de esto; simplemente pondremos nuestro script en el directorio para los CGI, y cuando se llame al CGI, el servidor se encargará de ejecutar el intérprete.

4.3.1 Especificaciones

Como un programa CGI es un ejecutable, es equivalente a dejar ejecutar un programa en tu sistema, que no es lo mas seguro a hacer. Por ello existen una serie de precauciones de seguridad que son necesarias de implementar cuando se usan programas CGI.

Probablemente la que afectará al usuario típico del Web, es que hecho de que los programas CGI necesitan residir en un directorio especial, así el servidor sabe que tiene que ejecutarlo, en vez de simplemente mostrarlo por pantalla. Este directorio está generalmente bajo el control del webmaster, prohibiendo al usuario medio crear programas CGI. Hay otros métodos para permitir el accesos a scripts CGI, pero depende del webmaster que se te de esta posibilidad.

Si dispones de una versión del servidor HTTPd NCSA, verás un directorio denominado /cgi-bin. Este es el directorio especial antes mencionado, donde todos los programas CGI residen. Un programa CGI se puede escribir en cualquier lenguaje que permita ser ejecutado en el sistema, como:

- C/C++
- Fortran
- PERL

- TCL
- Algún Shell de Unix
- Visual Basic

Simplemente depende de lo que tengas en tu sistema. Si usas un lenguaje de programación como C o Fortran, como ya sabrás, debes compilar el programa antes de poder ejecutarlo.

Si miras en el directorio /cgi-src, encontrarás el código fuente de algunos programas CGI del directorio /cgi-bin. Pero, si usas alguno de los lenguajes in terpretados, como PERL, TCL, o un shell de Unix, el script simplemente necesita residir en el directorio /cgi-bin, ya que no tiene un código fuente asociado. Mucha gente prefiere escribir scripts CGI en vez de programas, ya que son más fáciles de depurar, modificar y mantener que un programa típico compilado.

4.3.2 ¿Qué es el directorio cgi-bin?

Este es un directorio especial que contiene los scripts, configurado dentro del servidor http. El servidor conoce que este directorio contiene ejecutables que deberán ser ejecutados y su salida deberá ser enviada al navegador del cliente. No se puede simplemente crear un directorio cgi-bin, el administrador del servidor deberá configurarlo para su uso. Si no está configurado, los scripst serán cargados como simples ficheros de texto.

Algunos servidores están configurados de tal manera que los ficheros con una determinada extensión son reconocidos como scripts y serán ejecutados como si estuvieran en un directorio cgi-bin.

La configuración de los directorios, o de la extensión mencionada antes, depende únicamente del servidor. Comprueba la documentación sobre tu servidor, o pregunta a otro usuario que también lo use.

CAPÍTULO 5

5. SERVIDORES WEB - HERRAMIENTAS DE SEGURIDAD

A continuación una serie de herramientas, las mismas que nos ayudarán a proteger nuestro sistema. Para conseguirlo, tenemos dos tipos de herramientas. Las primeras, se basan en chequeos a los archivos. Las segundas, nos alertan de posibles modificaciones de archivos y de programas "sospechosos" que puedan estar ejecutándose en la máquina de forma camuflada. En primer lugar, las que chequean la integridad de los sistemas de archivos.

5.1 COPS (Computer Oracle and Password System)

Cops es un conjunto de programas diseñado por la Universidad de Purdue que chequea ciertos aspectos del sistema operativo UNIX relacionados con la seguridad. Existen dos versiones de este paquete: una versión escrita en "sh" y "C" y otra versión escrita en "perl", aunque su funcionalidad es similar.

Este programa es fácil de instalar y configurar y se ejecuta en gran cantidad de plataformas UNIX. En el primer caso, necesitaremos un compilador de lenguaje C y un shell estándar (sh). En el segundo, nos bastará con tener instalado el interprete de perl (versión 3.18 o superior).

Entre las funcionalidades que tiene Cops podemos destacar.

- Chequeo de modos y permisos de los archivos, directorios y dispositivos
- Passwords pobres. En el caso que tengamos una herramienta como crack, podemos comentar la línea de chequeo de passwords.
- Chequeo de contenido, formato y seguridad de los archivos de "password" y "group".
- Chequeo de programas con root-SUID.
- Permisos de escritura sobre algunos archivos de usuario como ".profile" y ".cshrc"
- Configuración de ftp "anonymous".
- Chequeo de algunos archivos del sistema como "hosts.equiv", montajes de NFS sin restricciones, "ftpusers", etc.

Veamos un ejemplo del archivo creado por este programa:

```
ATTENTION:
Security Report for Tue Apr 11 13:33:33 WET DST 1995 from host acme
Warning!  Root does not own the following file(s): /dev /usr/etc
Warning!  /etc/ethers is _World_ writable!
Warning!  File /etc/motd (in /etc/rc.local.orig) is _World_ writable!
Warning!  Password file, line 12, user sysdiag has uid = 0 and is not root
               sysdiag:*:0:1:Old
System
Diagnostic:/usr/diag/sysdiag:/usr/diag/sysdiag/sysdiag
Warning!  Password file, line 13, user sundiag has uid = 0 and is not root
               sundiag:*:0:1:System
Diagnostic:/usr/diag/sundiag:/usr/diag/sundiag/sundiag
Warning!  YPassword file, line 3, user sysdiag has uid = 0 and is not root
               sysdiag:*:0:1:Old
System
Diagnostic:/usr/diag/sysdiag:/usr/diag/sysdiag/sysdiag
Warning!  /etc/ftpusers should exist!
```

Figura 5. 1 Ejemplo del archivo de trazas generado por COPS

5.2 Tiger

Es un software desarrollado por la Universidad de Texas que está formado por un conjunto de shell scripts y código C que chequean el sistema para detectar problemas de seguridad de forma parecida a COPS.

Una vez chequeado el sistema, se genera un archivo con toda la información recogida por el programa. Tiger dispone de una herramienta que recibe como parámetro dicho archivo y da una serie de explicaciones adicionales de cada línea que generó el programa anterior. El programa viene con un archivo de configuración donde es posible informarle qué tipo de chequeo se quiere realizar.

Podemos comentar las operaciones más lentas y ejecutar éstas de forma menos continuada, mientras que las más rápidas pueden ser ejecutadas más frecuentemente.

Entre la información que chequea el programa tenemos:

- Configuración del sistema.
- Sistemas de archivos.
- Archivos de configuración de usuario.
- Chequeo de caminos de búsqueda.
- Chequeos de cuentas.
- Chequeos de alias.

- Comprueba la configuración de ftp "anonymous".
- Chequeo scripts de cron.
- NFS.
- Chequeo de servicios en el archivo /etc/inetd.conf
- Chequeo de algunos archivos de usuario (.netrc, .rhosts, .profile, etc)
- Comprobación archivos binarios (firmas). Para poder chequear éstos es necesario disponer de un archivo de firmas.

5.3 WebCrack

Este paquete de dominio público realizado por Alex Muffet permite chequear el archivo de contraseñas de UNIX y encontrar passwords triviales o poco seguras.

Para ello, usa el algoritmo de cifrado (DES) utilizado por el sistema UNIX y va comprobando a partir de reglas y de diccionarios las passwords que se encuentran en el archivo de contraseñas, creando un archivo con todos los usuarios y palabras descubiertas. Se realiza una serie de pasadas sobre el archivo de contraseñas, aplicando la secuencia de reglas que se especifique.

Estas reglas se encuentran en dos archivos (gecos.rules y dicts.rules) y pueden ser modificadas utilizando un lenguaje bastante simple. Para una mayor efectividad pueden utilizarse diccionarios complementarios (existen

en gran diversidad servidores ftp) en diferentes idiomas y sobre diversos temas.

Experiencias realizadas en la Universidad Carlos III de Madrid sobre diversas máquinas han arrojado resultados de 16% de passwords triviales en máquinas donde no se tenía ninguna norma a la hora de poner contraseñas de usuario.

Es una buena norma pasar de forma periódica el crack para detectar contraseñas poco seguras, además de tener una serie de normas sobre passwords, tanto en su contenido como en la periodicidad con que deben ser cambiadas.

5.4 Tripwire

Herramienta de comprobación de integridad de archivos. Tripwire ayuda a administradores y usuarios de sistemas monitoreando alguna posible modificación en algún set de archivos. Si se usa regularmente en los archivos de sistema (por Ej. diariamente), Tripwire puede notificar a los administradores del sistema, si algún archivo fue modificado o reemplazado, para que se puedan tomar medidas de control de daños a tiempo.

Ayuda a asegurar la integridad de ficheros y directorios de sistema esenciales identificando todos los cambios hechos a ellos. El uso de Tripwire para detectar intrusiones y fijar daños ayuda a mantenerlo al

tanto de los cambios del sistema y puede agilizar el restablecimiento de una entrada forzada reduciendo el número de ficheros que hay que restablecer para reparar el sistema.

Compara los ficheros y directorios con una base de datos de la ubicación de los ficheros, las fechas en que han sido modificados y otros datos. Tripwire genera la base tomando una instantánea de ficheros y directorios específicos en estado conocido como seguro. (Para máxima seguridad, Tripwire debería ser instalado y la base debería ser creada antes que el sistema sea expuesto al riesgo de intrusión).

Después de haber creado la base de datos, Tripwire compara el sistema actual con la base y proporciona información sobre cualquier modificación, añadidura, o supresión.

La base datos está compuesta por una serie de datos como la fecha de la última modificación, propietario, permisos, etc. con todo ello se crea una firma para cada archivo en la base de datos.

Esta herramienta debería ser ejecutada después de la instalación de la máquina con el objeto de tener una "foto" de los sistemas de archivos en ese momento y puede ser actualizada cada vez que añadimos algo nuevo. Dispone de un archivo de configuración que permite decidir qué parte del sistema de archivos va a ser introducida en la base de datos para su posterior comprobación.

5.5 CPM (Check Promiscuous Mode)

Este pequeño programa realizado por la Universidad de Carnegie Mellon, chequea la interfaz de red de la máquina descubriendo si está siendo utilizada en modo promiscuo (escuchando todo el tráfico de la red).

Está herramienta es muy útil, porque nos alerta de la posible existencia de un "sniffer" (olfateador) que intente capturar información en nuestra red como puedan ser las passwords. Este programa debería ser ejecutado de forma periódica para detectar lo antes posible el estado promiscuo en la placa de red. Una forma útil de utilizarlo es mandarnos el resultado vía correo electrónico.

Es importante tener en cuenta que muchos de los programas descritos en este documento, pueden poner la placa en modo promiscuo con lo que deberemos asegurarnos que no son nuestros programas los que producen esa alerta. Generalmente los programas tipo "sniffer" suelen estar ejecutándose como procesos camuflados en el sistema.

5.6 OSH (Operator Shell)

Creado por Mike Neuman, OSH es un software de dominio público es una shell restringida con "setuid root", que permite indicar al administrador mediante un archivo de datos qué comandos puede ejecutar cada usuario.

El archivo de permisos está formado por nombres de usuario y una lista de los comandos que se permite a cada uno de ellos. También es posible especificar comandos comunes a todos ellos.

Este shell deja una auditoría de todos los comandos ejecutados por el usuario, indicando si pudo o no ejecutarlos. Dispone, además, de un editor (vi) restringido.

Este programa es de gran utilidad para aquellas máquinas que dispongan de una gran cantidad de usuarios y no necesiten ejecutar muchos comandos, o para dar privilegios a determinados usuarios "especiales" que tengan algún comando que en circunstancias normales no podrían con un shell normal.

Veamos un ejemplo del logístico creado por el programa:

```
LOGIN: acme ran osh at Wed Jun 7 12:09:09 1995
acme    (6/7/95 12:09:11)pwd                    - acme    (6/7/95
12:09:13)ls              +
acme    (6/7/95          12:09:16) ls -la                  +
acme    (6/7/95 12:09:20)elm           -
acme    (6/7/95          12:09:23)quit           - acme    (6/7/95
12:09:27)exit            -
acme    (6/7/95 12:09:30)logout                 - acme    (6/7/95
12:09:33)exit            -
logout: acme left osh at Wed Jun 7 12:09:34 1995
```

Figura 5. 2 Ejemplo del archivo de trazas generado por OSH

5.7 Noshell

Este programa permite al administrador obtener información adicional sobre intentos de conexión a cuentas canceladas en una máquina.

Para utilizarlo basta sustituir el shell del usuario en el archivo /etc/password por éste programa. A partir de ahí, cada intento de conexión generará un mensaje (vía e- mail o syslog) indicando: usuario remoto, nombre de la computadora remota, dirección IP, día y hora del intento de login y tty utilizado para la conexión. Todas estas herramientas se pueden bajar de lince.uc3m.es o de cualquier sunsite.

5.8 Trinux

Trinux contiene las más populares herramientas de seguridad en redes y es usado para mapear y monitorear redes TCP/IP.

El paquete es muy interesante pues, básicamente, se compone varios discos, con los cuales se bootea la máquina que se va a dedicar a realizar el trabajo y corre enteramente en RAM.

Las aplicaciones que trae, principalmente, son:

- mail -soporte simple de correo saliente usando smail.
- netbase - utilitarios estándar de redes, tales como ifconfig, arp, ping, etc.
- netmap - herramientas de escaneo de red, tal como fyodor's, strobe, nmap y netcat.

- netmon - herramientas de monitoreo y sniffers, tal como sniffit, tcpdump y iptraf
- perlbase - base del lenguaje Perl.
- perli386 - archivos del sistema Perl.
- perlmods - módulos de Perl.
- pcmcia - soportes de módulos de kernel y scripts para laptop
- snmp - herramientas seleccionadas desde CMU SNMP.
- web - cliente Lynux.

5.9 Módulo mod_ssl

El módulo mod_ssl es un módulo de seguridad para el Servidor Web Apache. El módulo mod_ssl usa las herramientas suministradas por el OpenSSL Project para añadir una característica muy importante al Apache, la posibilidad de encriptar las comunicaciones.

A diferencia de las comunicaciones entre un navegador y un servidor web usando HTTP "normal", en la que se envía el texto íntegro, pudiendo ser interceptado y leído a lo largo del camino entre servidor y navegador.

5.10 El OpenSSL Project

Incluye un kit de herramientas que implementa los protocolos SSL (Secure Sockets Layer) y TLS (Transport Layer Security), así como una librería de codificación de propósito general.

El protocolo SSL se usa actualmente para la transmisión de datos segura sobre Internet; el protocolo TLS es un estándar de Internet para comunicaciones privadas (seguras)y fiables a través de Internet. Las herramientas OpenSSL son usadas por el módulo mod_ssl para aportar seguridad en las comunicaciones Web.

Requerimientos Recomendados

- Procesador: Pentium-class 200 MHz o superior
- RAM: 192MB para modo gráfico
- Disco Duro: 2.5GB de espacio; 4.5GB para instalación completa
- Monitor: SVGA (1028x1024) para ambiente gráfico
- CD –ROM: 32x con auto inicialización

CAPÍTULO 6

6. EVALUACIÓN DE NIVELES DE RIESGO DE LOS RECURSOS

6.1 Riesgos

La autenticación suele realizarse mediante una contraseña, aún cuando sería más lógico - si bien los costos resultan todavía altos para la mayoría de sistemas - que se pudiera combinar con características biométricas del usuario para impedir la suplantación.

Entre éstas pueden estar: la realización de la firma con reconocimiento automático por ordenador, el análisis del fondo de ojo, la huella digital u otras.

Al margen de la seguridad, nos parece que el mayor riesgo, aún teniendo un entorno muy seguro, es que la Informática y la Tecnología de la Información en general no cubran las necesidades de la entidad; o que no estén alineadas con las finalidades de la organización.

Limitándonos a la seguridad propiamente dicha, los riesgos pueden ser múltiples. El primer paso es conocerlos y el segundo es tomar decisiones al respecto; conocerlos y no tomar decisiones no tiene sentido y debiera crearnos una situación de desasosiego.

Dado que las medidas tienen un costo, a veces, los funcionarios se preguntan cuál es el riesgo máximo que podría soportar su organización. La respuesta no es fácil porque depende de la criticidad del sector y de la entidad misma, de su dependencia respecto de la información, y del impacto que su no disponibilidad pudiera tener en la entidad. Si nos basamos en el impacto nunca debería aceptarse un riesgo que pudiera llegar a poner en peligro la propia continuidad de la entidad, pero este listón es demasiado alto.

Por debajo de ello hay daños de menores consecuencias, siendo los errores y omisiones la causa más frecuente - normalmente de poco impacto pero frecuencia muy alta - y otros, como por ejemplo:

- El acceso indebido a los datos (a veces a través de redes),
- La cesión no autorizada de soportes magnéticos con información crítica (algunos dicen "sensible"),
- Los daños por fuego, por agua (del exterior como puede ser una inundación, o por una tubería interior),
- La variación no autorizada de programas, su copia indebida, y tantos otros, persiguiendo el propio beneficio o causar un daño, a veces por venganza.

Otra figura es la del "hacker", que intenta acceder a los sistemas sobre todo para demostrar (a veces, para demostrarse a sí mismo/a) qué es capaz de hacer, al superar las barreras de protección que se hayan establecido.

Alguien podría preguntarse por qué no se citan los virus, cuando han tenido tanta incidencia. Afortunadamente, este riesgo es menor en la actualidad comparando con años atrás. Existe, de todas maneras, un riesgo constante porque de forma continua aparecen nuevas modalidades, que no son detectadas por los programas antivirus hasta que las nuevas versiones los contemplan.

Un riesgo adicional es que los virus pueden llegar a afectar a los grandes sistemas, sobre todo a través de las redes, pero esto es realmente difícil - no nos atrevemos a decir que imposible- por las características y la complejidad de los grandes equipos y debido a las características de diseño de sus sistemas operativos.

En definitiva, las amenazas hechas realidad pueden llegar a afectar los datos, en las personas, en los programas, en los equipos, en la red y algunas veces, simultáneamente en varios de ellos, como lo puede ser un incendio.

Podríamos hacernos una pregunta realmente difícil: ¿qué es lo más crítico que debería protegerse? La respuesta de la mayoría, probablemente, sería que las personas resultan el punto más crítico y el valor de una vida humana no se puede comparar con las computadoras, las aplicaciones o los datos de cualquier entidad. Ahora bien, por otra parte, podemos determinar que los datos son aún más críticos si nos centramos en la continuidad de la entidad.

Como consecuencia de cualquier incidencia, se pueden producir unas pérdidas que pueden ser no sólo directas (comúnmente que son cubiertas por los seguros) más fácilmente, sino también indirectas, como la no recuperación de deudas al perder los datos, o no poder tomar las decisiones adecuadas en el momento oportuno por carecer de información.

Sabemos que se producen casos similares en gran parte de entidades, pero en general no conocemos a cuáles han afectado (o lo sabemos pero no podemos difundirlo), porque por imagen estos no se hacen públicos y el hecho de que se conozcan muchos más referidos a Estados Unidos y a otros puntos lejanos que respecto de nuestros países no significa que estemos a salvo, sino que nuestro pudor es mayor y los ocultamos siempre que podemos.

6.2 ¿Cómo establecer los niveles de riesgo de los recursos involucrados?

Al crear una política de seguridad de red, es importante entender que la razón para crear tal política es, en primer lugar, asegurar que los esfuerzos invertidos en la seguridad son costeables.

Esto significa que se debe entender cuáles recursos de la red vale la pena proteger y que algunos recursos son más importantes que otros. También se deberá identificar la fuente de amenaza de la que se protege a los recursos. A pesar de la cantidad de publicidad sobre intrusos en una red, varias encuestas indican que para la mayoría de las organizaciones, la

pérdida real que proviene de los "miembros internos" es mucho mayor (tal cual se ha explicado anteriormente).

El análisis de riesgos implica determinar lo siguiente:

- Qué se necesita proteger
- De quién protegerlo
- Cómo protegerlo

Los riesgos se clasifican por el nivel de importancia y por la severidad de la pérdida. No se debe llegar a una situación donde se gasta más para proteger aquello que es menos valioso.

En el análisis de los riesgos, es necesario determinar los siguientes factores:

- Estimación del riesgo de pérdida del recurso (lo llamaremos Ri)
- Estimación de la importancia del recurso (lo llamaremos Wi)

Como un paso hacia la cuantificación del riesgo de perder un recurso, es posible asignar un valor numérico. Por ejemplo, al riesgo (Ri) de perder un recurso, se le asigna un valor de cero a diez, donde cero, significa que no hay riesgo y diez es el riesgo más alto.

De manera similar, a la importancia de un recurso (Wi) también se le puede asignar un valor de cero a diez, donde cero significa que no tiene importancia y diez es la importancia más alta. La evaluación general del

riesgo será entonces el producto del valor del riesgo y su importancia (también llamado el peso). Esto puede escribirse como:

$$WR_i = R_i * W_i$$

Donde:

WRi : es el peso del riesgo del recurso "i" (también lo podemos llamar ponderación)

Ri : es el riesgo del recurso "i"

Wi : es la importancia del recurso "i"

Los recursos que deben ser considerados al estimar las amenazas a la seguridad son solamente seis:

Hardware: procesadores, tarjetas, teclados, terminales, estaciones de trabajo, computadoras personales, impresoras, unidades de disco, líneas de comunicación, cableado de la red, servidores de terminal, routers, bridges.

Software: programas fuente, programas objeto, utilerías, programas de diagnóstico, sistemas operativos, programas de comunicaciones.

Datos: durante la ejecución, almacenados en línea, archivados fuera de línea, back-up, bases de datos, en tránsito sobre medios de comunicación.

Gente: usuarios, personas para operar los sistemas.

El otro problema que nos presentamos, es el de las intromisiones clandestinas.

Aquí, es preciso tener en cuenta el tipo de recurso a proteger. En base a ello, estará dada la política de seguridad.

Daremos, a continuación, algunos ejemplos acerca de lo qué tenemos que enfrentarnos:

- ¿Cómo aseguramos que no están ingresando a nuestro sistema por un puerto desprotegido o mal configurado?
- ¿Cómo nos aseguramos de que no se estén usando programas propios del sistema operativo o aplicaciones para ingresar al sistema en forma clandestina?
- ¿Cómo aseguramos de que, ante un corte de energía eléctrica, el sistema seguirá funcionando?
- ¿Cómo nos aseguramos de que los medios de transmisión de información no son susceptibles de ser monitoreados?
- ¿Cómo actúa la organización frente al alejamiento de uno de sus integrantes?

La respuesta a estos interrogantes reside en la posibilidad de conseguir dicha seguridad por medio de herramientas de control y seguimiento de accesos, utilizando check-lists para comprobar puntos importantes en la configuración y/o funcionamiento de los sistemas y por medio de procedimientos que hacen frente a las distintas situaciones.

Es muy aconsejable que se disponga de una agenda con las tareas que se deben llevar a cabo regularmente, a fin de que el seguimiento de los datos obtenidos sea efectivo y se puedan realizar comparaciones válidas al contar con datos secuenciales.

Esta agenda, podría ser en sí misma un procedimiento.

Damos, a continuación, un ejemplo de procedimiento de chequeo de eventos en el sistema:

Diariamente:

- Extraer un logístico sobre e' volumen de correo transportado.
- Extraer un logístico sobre las conexiones de red levantadas en las últimas 24 horas.

Semanalmente:

- Extraer un logístico sobre los ingresos desde el exterior a la red interna.
- Extraer un logístico con las conexiones externas realizadas desde nuestra red.
- Obtener un logístico sobre los downloads de archivos realizados y quién los realizó.
- Obtener gráficos sobre tráfico en la red.
- Obtener logísticos sobre conexiones realizadas en horarios no normales (desde dónde, a qué hora y con qué destino).

Mensualmente:

- Realizar un seguimiento de todos los archivos logísticos a fin de detectar cambios (realizados con los archivos de back-up del mes anterior).

Cabría resaltar que, en gran parte, este procedimiento puede ser automatizado por medio de programas que realicen las tareas y sólo informen de las desviaciones con respecto a las reglas dadas.

6.3 Evaluación de Riesgos

El análisis de riesgos supone más que el hecho de calcular la posibilidad de que ocurran cosas negativas.

- Se debe poder obtener una evaluación económica del impacto de estos sucesos. Este valor se podrá utilizar para contrastar el costo de la protección de la información en análisis, versus el costo de volverla a producir.
- Se debe tener en cuenta la probabilidad que sucedan cada uno de los problemas posibles. De esta forma se pueden priorizar los problemas y su coste potencial desarrollando un plan de acción adecuado.
- Se debe conocer qué se quiere proteger, dónde y cómo, asegurando que con los costos en los que se incurren se obtengan beneficios efectivos. Para esto se deberá identificar los recursos

(hardware, software, información, personal, accesorios, etc.) con que se cuenta y las amenazas a las que se está expuesto.

La evaluación de riesgos y presentación de respuestas debe prepararse de forma personalizada para cada organización; pero se puede presupone algunas preguntas que ayudan en la identificación de lo anteriormente expuesto.

- ¿Que puede ir mal?
- ¿Con qué frecuencia puede ocurrir?
- ¿Cuáles serían sus consecuencias?
- ¿Qué fiabilidad tienen las respuestas a las tres primeras preguntas?
- ¿Se está preparado para abrir las puertas del negocio sin sistemas, por un día, una semana, cuanto tiempo?
- ¿Cuál es el costo de una hora sin procesar, un día, una semana...?
- ¿Cuánto, tiempo se puede estar fuera de línea sin que los clientes se vayan a la competencia?
- ¿Se tiene forma de detectar a un empleado deshonesto en el sistema?
- ¿Se tiene control sobre las operaciones de los distintos sistemas?
- ¿Cuántas personas dentro de la empresa, (sin considerar su honestidad), están en condiciones de inhibir el procesamiento de datos?
- ¿A que se llama información confidencial y/o sensitiva?

- ¿La seguridad actual cubre los tipos de ataques existentes y está preparada para adecuarse a los avances tecnológicos esperados?
- ¿A quién se le permite usar que recurso?
- ¿Cuáles serán los privilegios y responsabilidades del Administrador y las del usuario?
- ¿Cómo se actuará si la seguridad es violada?

Tipo de Riesgo	Factor
Robo de Hardware	Alto
Virus Informáticos	Alto
Vandalismo	Medio
Fallas en los equipos	Medio
Accesos no autorizados	Medio
Equivocaciones	Medio
Fraude	Bajo
Fuego	Muy Bajo
Terremotos	Muy Bajo

Tabla 6.1 Ejemplo de Tipo de Riesgo y su factor de incidencias

Según esta tabla habrá que tomar las medidas pertinentes de seguridad para cada caso en particular, cuidando incurrir en los costos necesarios según el factor de riesgo representado.

6.4 Niveles de Riesgos

Los riesgos se clasifican por su nivel de importancia y por la severidad de su pérdida:

1. Estimación del riesgo de pérdida del recurso (Ri)

2. Estimación de la importancia del Recurso (Ii)

Para la cuantificación del riesgo de perder un recurso, es posible asignar un valor numérico de 0 a 10, tanto a la importancia del recurso (10 es el recurso de mayor importancia) como al riesgo de perderlo (10 es el riesgo mas alto).

El riesgo de un recurso será el producto de su importancia por el riesgo de perderlo elevado a la 7ª potencia.

$$WR_i = R_i * I_i$$

Luego, con la siguiente fórmula es posible calcular el riesgo general de los recursos de la red:

$$W_R = \frac{(WR_1 * I_1 + WR_2 * I_2 + \ldots + WR_n * I_n)}{I_1 + I_2 + \ldots + I_n}$$

Otros factores que debe considerar para el análisis de riesgo de un recurso de red son su disponibilidad, su integridad y su carácter confidencial, los cuales pueden incorporarse a la fórmula para ser evaluados.

Ejemplo: el Administrador de una red ha estimado los siguientes riesgos y sus importancias para los elementos de la red que administra:

Recurso	Riesgo (Ri)	Importancia (Ii)	Riesgo Evaluado (Ri * Ii)
Router	6	7	42
Gateway	6	5	30
Servidor	10	10	100
PC's	9	2	18

Tabla 6.2 Valuación de Riesgos

El recurso que más debe protegerse es el servidor. Para la obtención del riesgo total de la red calculamos:

$$W_R = \frac{42 + 30 + 100 + 18}{7 + 5 + 10 + 2} = 7.92$$

6.5 Identificación de Amenaza

Una vez conocidos los riesgos, los recursos que se deben proteger y como su daño o falta pueden influir en la organización es necesario identificar cada una de las amenazas y vulnerabilidades que pueden causar estas bajas en los recursos. Como ya se mencionó existe una relación directa entre amenaza y vulnerabilidad a tal punto que si una no existe la otra tampoco. Se suele dividir las amenazas existentes según su ámbito de acción:

- Desastre del entorno (Seguridad física)
- Amenazas del sistema (Seguridad Lógica)
- Amenazas en la red (Comunicaciones)
- Amenazas de personas (Insiders – Outsiders)

Se debería disponer de una lista de amenazas (actualizadas) para ayudar a los administradores de seguridad a identificas los distintos métodos, herramientas y técnicas de ataque que se pueden utilizar.

Es importante que los Administradores actualicen constantemente sus conocimientos en esta área, ya que los nuevos métodos, herramientas y técnicas para sortear las medidas de seguridad evolucionan de forma continua.

Es necesario tener una metodología para definir una estrategia de seguridad informática que se puede utilizar para implementar directivas y controles de seguridad con el objeto de aminorar los posibles ataques y amenazas.

La metodología se basa en los distintos ejemplos (uno para cada tipo de amenaza) y contempla como hubiera ayudado una política de seguridad en caso de haber existido.

6.6 Evaluación de Costos

Desde un punto de vista oficial, el desafío de responder la pregunta del valor de la información ha sido siempre difícil, y más difícil aún hacer estos costos justificables, siguiendo el principio que "si desea justificarlos, debe darle un valor" elevado a la 8ª potencia.

Establecer el valor de los datos es algo totalmente relativo, pues la información constituye un recurso que, en muchos casos, no se valora

adecuadamente debido a su intangibilidad, cosa que no ocurre con los equipos, la documentación o las aplicaciones.

Además, las medidas de seguridad no influyen en la productividad del sistema por lo que las organizaciones son reticentes a dedicar recursos a esta tarea. Por eso es importante entender que los esfuerzos invertidos en la seguridad son costeables.

La evaluación de costos más ampliamente aceptada consiste en cuantificar los daños que cada posible vulnerabilidad puede causar teniendo en cuenta las posibilidades. Un planteamiento posible para desarrollar esta política es el análisis de lo siguiente:

- ¿Qué recursos se quieren proteger?
- ¿De qué personas necesita proteger los recursos?
- ¿Qué tan reales son las amenazas?
- ¿Qué tan importante es el recurso?
- ¿Qué medidas se pueden implantar para proteger sus bienes de una manera económica y oportuna?

Con esas sencillas preguntas (más la evaluación de riesgo) se debería conocer cuáles recursos vale la pena (y justifican su costo) proteger, y entender que algunos son más importantes que otros.

El objetivo que se persigue es log·ar que un ataque a los bienes sea más costoso que su valor, invirtiendo menos de lo que vale. Para esto se define tres costos fundamentales.

- **CP:** Valor de los bienes y recursos protegidos
- **CR:** Costo de los medios necesarios para romper las medidas de seguridad establecidas.
- **CS:** Costo de las medidas de seguridad.

Para que la política de seguridad sea lógica y consistente se debe cumplir que:

- **CR > CP:** o sea que un ataque para obtener los bienes debe ser más costoso que el valor de los mismos. Los beneficios obtenidos de romper las medidas de seguridad no deben compensar el costo de desarrollo del ataque.
- **CP > CS:** o sea que el costo de los bienes protegidos debe ser mayor que el costo de la protección.

Luego **CR > CP > CS** y lo que se busca es:

- **Minimizar:** el costo de la protección manteniéndolo por debajo de los bienes protegidos. Si proteger los bienes es más caro de lo que valen (el lápiz dentro de la caja fuerte), entonces resulta mas conveniente obtenerlos de nuevo en vez de protegerlo.

- **Maximizar:** el costo de los ataques manteniéndolo por encima del de los bienes protegidos. Si atacar el bien es más caro de lo que valen, al atacante le conviene más obtenerlo de otra forma menos costosa.

Se debe tratar de valorar los costos en que se puede incurrir en el peor de los casos contrastando con el costo de las medidas de seguridad adoptadas. Se debe poner especial énfasis en esta etapa para no incurrir en el error de no considerar costos, muchas veces, ocultos y no obvios (costos derivados).

6.6.1 Valor Intrínseco

Es el más fácil de calcular (pero no fácil) ya que solo consiste en otorgar un valor a la información contestando preguntas como las mencionadas y examinando minuciosamente todos los componentes a proteger.

6.6.2 Costos derivados de la pérdida

Una vez más deben abarcarse todas las posibilidades, intentando descubrir todos los valores derivados de la pérdida de algún componente del sistema.

Muchas veces se trata del valor añadido que gana un atacante y la repercusión de esa ganancia para el entorno, además de costo del elemento perdido. Deben considerarse elementos como:

Información aparentemente inocua como datos personales, que pueden permitir a alguien suplantar identidades. Datos confidenciales de acuerdos y contratos que un atacante podría usar para su beneficio.

Tiempos necesarios para obtener ciertos bienes. Un atacante podría acceder a ellos para ahorrarse el costo y tiempo necesario para su desarrollo.

6.6.3 Punto de equilibrio

Una vez evaluados los riesgos y los costos en los que se está dispuesto a incurrir y decidido el nivel de seguridad a adoptar, podrá obtenerse un punto de equilibrio entre estas magnitudes.

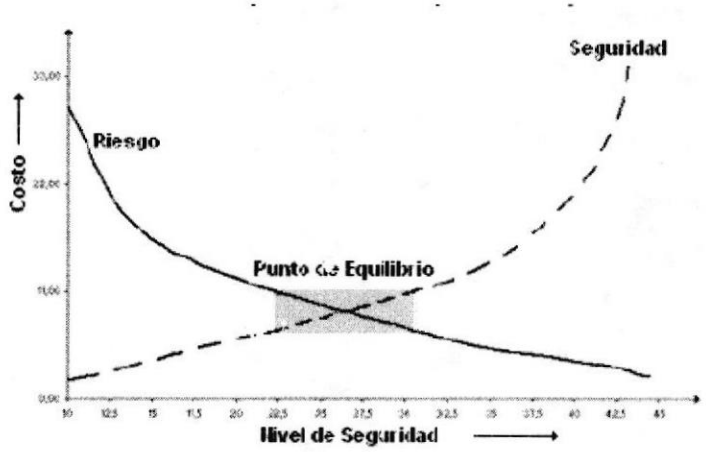


Figura 6.1 Punto de Equilibrio: Costo / Seguridad / Riesgo

Como puede apreciarse los riesgos disminuyen al aumentar la seguridad (y los costos en los que incurre) pero como ya se sabe los costos tenderán al infinito sin lograr el 100% de seguridad y por supuesto

nunca se logrará no correr algún tipo de riesgo. Lo importante es lograr conocer cuan seguro se estará conociendo los costos y los riesgos que se corren (Punto de Equilibrio).

CAPÍTULO 7

7. SERVIDORES WEB – PROTECCIÓN ADICIONAL USANDO FIREWALL

7.1 Firewalls

7.1.1 Definición

"Un Firewall es un sistema o grupo de sistemas que establece una política de control de acceso entre dos redes"

Tienen las siguientes propiedades:

- Todo el trafico de adentro hacia fuera, y viceversa debe pasar a través de ella.
- Solo el tráfico autorizado, definido por la política de seguridad es autorizado para pasar por él.
- El sistema es realmente resistente a la penetración.

7.1.2 Tráfico en Internet

Cuando nos referimos a que todo el trafico de adentro hacia afuera y viceversa, debe pasar por un Firewall, esto es con respecto al protocolo TCP/IP.

Para controlar trafico de TCP/IP se debe tener una clara idea de cómo trabaja, un protocolo es una descripción formal de cómo de los mensajes que serán intercambiados y las reglas que deben seguir dos o mas sistemas para transferirlos de tal forma que ambos puedan entenderse.

TCP (Transmission Control Protocol), divide los datos en partes, estas partes son llamadas paquetes, y le da a cada uno un número. Estos paquetes pueden representar texto, gráficas, sonido o vídeo ¾ cualquier elemento que la red pueda transmitir. La secuencia de números ayuda a asegurar que los paquetes puedan ser re ensamblados una vez recibidos.

Entonces cada paquete consiste en contenido, o datos, y la información que el protocolo necesita para hacerlo funcionar, llamado protocolo encabezado.

7.1.3 Firewalls como filtros

El Router es un tipo especial de switch el cual realiza el trabajo de hacer las conexiones externas y convertir el protocolo IP a protocolos de WAN y LAN.

Los paquetes de datos transmitidos hacia Internet, desde un visualizador de una PC, pasarán a través de numerosos ruteadores a lo largo del camino, cada uno de los cuales toma la decisión de hacia donde dirigir el trabajo.

Los ruteadores toman sus decisiones basándose en tablas de datos y reglas, por medio de filtros, por ejemplo, solo datos de una cierta dirección pueden pasar a través del ruteador, esto transforma un ruteador que puede filtrar paquetes en un dispositivo de control de acceso o Firewall.

Si el ruteador puede generar un registro de accesos esto lo convierte en valioso dispositivo de seguridad.

Si el servidor de Internet solicita información, o bien la suministra hacia sistemas de bases de datos distribuidas, entonces esta conexión entre el servidor y la estación de trabajo debería ser protegida.

7.1.4 Firewalls como Gateways

Las Firewalls son comúnmente referidas como Gateways, controlan el acceso desde afuera hacia adentro y viceversa. Un Gateway es una computadora que proporciona el servicio de intercambio de datos entre dos redes, sin embargo, un Firewall puede consistir en un poco más que un ruteador filtrador, este puede ser considerado como un Gateway controlado.

El tráfico va hacia la Gateway, en vez de dirigirse directamente hacia la red, la Gateway que pasa los datos, de acuerdo a la política de control de los accesos, a través de un filtro, hacia otra red o hacia otra Gateway conectada a otra red.

Esta mediación toma en cuenta, direcciones de fuente y destino, tipos de paquetes de datos, política de seguridad. Típicamente un Firewall registra los accesos y los intentos de acceso de una red a otra.

7.1.5 Firewalls como puntos de verificación

Algunas Firewalls proveen servicios de seguridad adicionales. Como encriptación y desencriptación, ambas deben usar sistemas compatibles de encriptación. Existen varios fabricantes que ofrecen dichos sistemas. Encriptación de Firewall a Firewall es la forma que se usa en el Internet de hoy.

Verificar la autenticidad del usuario así como el sistema que este usando también es importante, y los Firewalls pueden hacerlo, usando tarjetas inteligentes, fichas y otros métodos. Los Firewalls, pueden incluso proteger otras redes exteriores. Una compañía puede aplicar las mismas restricciones de tráfico, mejorad(con autenticación.

7.1.6 Firewalls internos

Alguien fuera de la empresa podría solicitar cierta información, pero no necesariamente necesita acceder a toda la información interna.

En estas circunstancias, los Firewalls juegan un importante papel forzando políticas de control de acceso entre redes confiables protegidas y redes que no son confiables.

En una WAN que debe ofrecer conexión de cualquier persona a cualquiera, otras formas en el nivel de aplicación pueden ser implementadas para proteger datos importantes.

Sin embargo, separar las redes por medio de Firewalls reduce significativamente los riesgos del ataque de un hacker desde adentro, esto es acceso no autorizado por usuarios autorizados. Agregando encriptación a los servicios del Firewall la convierte en una conexión Firewall a Firewall muy segura.

Esto siempre permite redes grandes interconectadas por medio de Internet. Agregando autenticación se puede aumentar el nivel de seguridad.

7.1.7 Tipos de Firewalls

Tanto se ha hablado de Firewalls desde el punto de vista de amenazas, principios y política. Ahora cambiamos a los específicos de implementaron de mecanismos que le permitan a las Firewalls la aplicación de políticas y proveer protección.

Cuando se habla de Firewalls, uno debería tomar en cuenta que la tecnología evoluciona muy rápidamente. Los Firewalls de hoy tienden a combinar diferentes mecanismos, haciendo difícil clasificarlos. Por esa razón se describen los ingredientes que pueden ir en el diseño de un Firewall.

7.1.8 Filtrado de Paquetes

Todos los Firewalls desempeñan algún tipo de filtrado de paquete IP, comúnmente por medio de un ruteador de filtrado de paquetes. El ruteador filtra paquetes, haciendo que ellos pasen por el ruteador, implementando un conjunto de reglas con base en la política de la Firewall. Un ruteador filtrador de paquetes, usualmente puede filtrar paquetes IP con base en algunos o todos los criterios siguientes:

- dirección fuente IP,
- dirección destino IP,
- puerto fuente TCP/UDP, y
- puerto destino TCP/UDP.

El filtrado puede bloquear conexiones desde o a las redes o anfitriones específicos, y pueden bloquear conexiones a puertos específicos. Un sitio podría desear bloquear las conexiones desde ciertas direcciones, tales como desde anfitriones o los sitios consideraron hostiles o indignos de confianza. Alternativamente, un sitio puede desear bloquear conexiones desde todos las direcciones externas al sitio (con ciertas excepciones, tales como con SMTP para recibir e-mail).

Los servidores tales como el TELNET DAEMON usualmente reside en puertos conocidos (puerto 23 para TELNET), así si un Firewall puede bloquear conexiones TCP o UDP desde puertos específicos, entonces el sitio puede hacer llamadas para asegurar los tipos de conexiones para

ser hechas a ciertos anfitriones pero no a otros. Por ejemplo, una compañía podría desear bloquear todas las conexiones de entrada a todos los host a excepción a algunos sistemas conexos de Firewall. A esos sistemas, quizás solo los servicios específicos serán permitidos, tal como SMTP para un sistema y conexiones TELNET o FTP a otro sistema (ven diagrama en la figura 5.1). Con filtrado sobre puertos TCP o UDP, esta política puede ser exitosa en este estilo de ruteador de filtrado de paquetes o un anfitrión con capacidad de filtrado de paquete.

El filtrado de paquetes en TELNET y SMTP [wack]. Un ejemplo básico para usar el filtrado de paquetes para implementar políticas pudiera ser permitir solo ciertas conexiones a una red de dirección 123.4.*.*.

Las conexiones TELNET serían permitidas a solo un host, 123.4.5.6, los cual podrían ser una aplicación Gateway en el sitio TELNET, y las conexiones SMTP serian permitirán a dos hosts, 123.4.5.7 y 123.4.5.8, que podrían ser dos sitios Gateway e-mail. NNTP (protocolo de noticias transfiere de red) es permitir solo desde el sitio de alimentación del sistema NNTP, 129.6.48.254, y solo al servidor NNTP del sitio, 123.4.5.9; NTP (el protocolo de tiempo de red) se permite a todos los host (anfitriones).

El ruteador de filtrado de paquetes bloqueará cualquiera otros servicios y paquetes. Este ejemplo muy básico de filtrado de paquete puede

volverse más complejo y flexible como el sitio fomenta ajustes a las reglas de filtrado.

Desafortunadamente, los ruteadores de filtrado de paquetes no pueden hacer todo. Ellos han sido tradicionalmente difíciles de usar en su configuración y mantenimiento. Esto está cambiando, con los vendedores poniendo más atención a las interfaces.

Las reglas de filtrado de paquetes son inherentemente complejas para especificar y usualmente no existe facilidad de prueba para averiguar la corrección de las reglas (a excepción de la prueba exhaustiva por handsee para probar). Además, algunos ruteadores no dan la capacidad para registrar (loggin), así que si las reglas de un ruteador permiten paquetes peligrosos, los paquetes pueden no ser detectados hasta que ocurra una caída del sistema. Los sitios que optan por usar ruteador de filtrado de paquetes para su Firewall deberán buscar por uno que ofrece registro (loggin) extensivo, una configuración simplificada y alguna forma de prueba de reglas.

Las excepciones a las reglas de filtración frecuentemente se necesitarán para permitir ciertos tipos de acceso que normalmente se bloquean. Esas excepciones pueden hacer las reglas de filtración tan complejo como ser inmanejables. Por ejemplo, es relativamente directo la especificación de una regla para bloquear todas las conexiones encaminadas al puerto 23 (el servidor TELNET), pero algunos sitios hacen excepciones para que

ciertos sistemas especificados puedan aceptar conexiones TELNET directamente. Para hacer esto, el administrador debe agregar una regla para cada sistema. (Algunos sistemas de filtración de paquetes adjuntan la importancia a la orden secuencial de las reglas de filtrado, permitiendo al administrador poner una excepción de permisos al sistema específico, seguido por una negación para todos los demás sistemas). La adición de ciertas reglas de esta manera puede complicar el sistema entero de filtración.

Algunos ruteadores de filtración de paquetes no filtra en los puertos fuente TCP/UDP, el cual puede hacer el filtrado más complejo el conjunto de reglas y puede abrir hoyos en el esquema de filtración.

7.1.9 Servidores Proxy

Un servidor proxy (algunas veces se hace referencia a el con el nombre de "Gateway" - puerta de comunicación - o "forwarder" - agente de transporte -), es una aplicación que media en el tráfico que se produce entre una red protegida e Internet. Los proxies se utilizan a menudo, como sustitutos de routers controladores de tráfico, para prevenir el tráfico que pasa directamente entre las redes. Muchos proxies contienen logines auxiliares y soportan la autentificación de usuarios. Un proxy debe entender el protocolo de la aplicación que está siendo usada, aunque también pueden implementar protocolos específicos de seguridad (por ejemplo: un proxy FTP puede ser configurado para permitir FTP entrante y bloquear FTP saliente).

Los servidores proxy, son aplicaciones específicas. Un conjunto muy conocido de servidores proxy son los TIS Internet Firewall Toolkit "FWTK", que incluyen proxies para Telnet, rlogin, FTP, X-Windows, http/Web, y NNTP/Usenet news. SOCKS es un sistema proxy genérico que puede ser compilado en una aplicación cliente para hacerla trabajar a través de una Firewall.

7.1.10 Aplicación Gateway

Para contar algunas de las debilidades asociado con el ruteador de filtrado de paquetes, los desarrolladores han creado aplicaciones de software que adelantan y filtran conexiones para servicios tal como telnet y ftp. Las aplicaciones referidas son servidores proxy, también conocido como aplicación Gateway. Las máquinas host corriendo los servidores proxy se refieren como Firewalls de aplicación Gateway. Trabajando junto, Firewalls de aplicación Gateway y el ruteador de filtrado de paquetes pueden potencialmente dar más altos niveles seguridad y flexibilidad que una sola. Por ejemplo, considera un sitio que bloquea todas las conexiones de Gateway TELNET y FTP que usan un ruteador de filtrado de paquetes permite a los paquetes de TELNET y FTP ir a un host solamente. Un usuario quien desea conectarse a través del sistema debe tener que conectar primero a la aplicación Gateway, y entonces al host de destino.

Los Firewalls de aplicación Gateway únicamente permiten esos servicios para cuales hay un proxy. En otras palabras, si un Gateway de aplicación

contiene proxy para FTP y TELNET, entonces solo FTP y TELNET puede permitirse en la subred protegida y todos los otros servicios son completamente bloqueados.

Para algunos sitios, este grado de seguridad es importante, como garantiza que solo los servicios considerados confiables se permiten mediante el Firewall. Esto también previene que otros servicios intrusos estén siendo implementar a espaldas de los administradores de Firewall.

Las aplicaciones Gateway ofrecen un número de ventajas generales sobre el modo default de permitir la que aplicación trafique directamente para host internos. Estas ventajas incluyen:

- Ocultamiento de la información. los nombres de sistemas internos no necesariamente necesitan ser conocidos por medio de DNS para los sistemas externos, desde la aplicación Gateway puede ser el host la única cuyo nombre debe hacerse conocido afuera de los sistemas.
- Robusta autenticación y registro. la Gateway puede autentificar el trafico de la aplicación antes este llegue a los host internos. El tránsito puede entonces ser registrado más efectivamente que con el registro estándar del host .
- Costo - eficacia. la tercera parte de software o hardware para la autenticación o registro necesario para ser ubicar solo en la aplicación Gateway.

- Menos complejas las reglas de filtrado. Las reglas en el ruteador de filtrado de paquetes serán menos complejas que aquellas que serían si el ruteador necesitara el filtrar el trafico de la aplicación y dirigir este a un número de sistemas específicos. El ruteador necesita solo permitir tráfico destinado para la aplicación Gateway y rechaza el resto.

7.1.11 Monitoreo de Paquetes

Algunos Firewalls de Internet combinan el filtrado de paquetes y el enfoque de aplicaciones Gateway, usando un filtrado de paquetes o un ruteador de hardware para controlar los niveles bajos de comunicación, y Gateway para habilitar aplicaciones. Esto pude crear un alto grado de control de acceso. Como siempre, este adaptamiento puede limitar en transparencia, flexibilidad y conectividad, y puede también dar una mayor dificultad en términos de configuración, manejo y especialización.

Otro punto de vista que gana aceptación es la inspección de paquetes que no solo los filtra, esto es, considerar su contenido tanto como sus direcciones.

Los Firewalls de este tipo emplean una inspección de módulos, aplicable a todos los protocolos que comprenden los datos de los paquetes destinados desde el nivel network (IP) hasta el nivel de aplicación. Esta estrategia puede proveer seguridad sensitiva al contexto para complejas aplicaciones y puede ser mas efectiva que la tecnología que solo tiene

acceso los datos en ciertos niveles. Por ejemplo las aplicaciones Gateway solo acceden a los datos de nivel aplicación, los ruteadores tienen acceso solo a niveles bajos, el enfoque de la inspección de paquetes integra toda la información reunida de todos los niveles en un simple punto de inspección.

Algunas Firewall de inspección también toman en cuenta el estado de la conexión, por ejemplo, la legítima entrada de paquetes puede ser probada con la petición de salida para ese paquete y se le permite entrar. Por el contrario, un paquete de entrada se enmascara con su respuesta a una inexistente petición de salida, este será bloqueado. Esto lleva el enfoque de tan llamado estado (stateful) mas allá del filtrado de paquetes. La inspección de módulos usa previas comunicaciones para derivar el estado actual de la comunicación que se esta realizando.

El filtrado inteligente puede efectivamente combinarse con la habilidad del rastreo de la sesión de red. Para usar la información acerca del inicio y fin de la sesión en la decisión de filtrado. Esto es conocido como filtrando sesión (sesión filtering). Los filtros usan reglas inteligentes, así aumenta el proceso de filtrado y controlando el rastreo de sesiones de la network que controla los paquetes individuales.

Una sesión de network contiene paquetes que van en dos direcciones, así que sin una sesión de filtrado cada sesión requiere dos reglas de filtrado de paquetes.

La primera controla los paquetes que van desde el originario host hasta el destinatario host. Una regla inteligente, sobre la otra mano, sabemos que regresando el paquete dirigidos en sentido opuesto y así no necesitamos la segunda regla.

Este enfoque ofrece ventajas considerables, desde los sitios que comúnmente tratan los paquetes originados afuera de la Firewall de manera diferente que los paquetes que regresan desde una conexión autorizada afuera.

7.1.12 Los Firewalls Híbridos

En la práctica, muchos de los Firewalls comerciales que hoy encontramos usan una combinación de estas técnicas.

Por ejemplo, un producto que se originó como un Firewall filtrador de paquetes, puede haber sido mejorado con filtrando inteligente a nivel de aplicación.

Las aplicaciones proxy en áreas establecidas como FTP pueden agregar una inspección de filtrado base en su esquema.

Nota: Recuerde, agregar los métodos de seguridad no significa necesariamente un aumento en la seguridad. Los mecanismos adicionales pueden aumentar, disminuir, o dejar infectado la postura de seguridad del Firewall.

7.1.13 Stealth Firewalls

7.1.13.1 Introducción

Los activos de una compañía entran en riesgo cuando ésta se conecta a Internet. Pero a pesar de tales peligros, las compañías desean alcanzar su objetivo de proveer sus productos y servicios a través de Internet a sus clientes y demás usuarios reduciendo los costos mientras que le permiten a sus empleados conectarse a la red para acceder la información corporativa o externa necesaria para desempeñar las labores en forma adecuada e incrementar la productividad de la organización.

Es por este motivo, que se adopcan tecnologías que permiten aumentar el nivel de confiabilidad y minimizar las pérdidas esperadas cuando se trata de salir al mundo de la red utilizando tecnologías como los denominados Firewalls. El objeto del presente análisis, radicará principalmente en el estudio de un importante grupo de estos. El de los Firewalls Stealth.

Actualmente, existen algunos tipos de comportamiento de los firewalls conocidos por la industria como comportamientos de tipo stealth. Técnicamente, el primero de ellos se encarga de bloquear el sistema (ya sea una estación de trabajo o un servidor) en presencia del tráfico de la red. Esto se lleva a cabo mediante la restricción de los servicios para poder utilizar únicamente el mínimo necesario y el uso de filtros de paquetes para controlar los recursos de red con los que se comunica. Un segundo tipo, es el firewall de red que no responde a peticiones de

recursos restringidos con mensajes como "resource denied" o "resource restricted", sino que simplemente ignora las peticiones hechas por otros nodos de la red. Este último tipo de comportamiento es también propio de un firewall de red, pero se diferencia en que no hace enrutamiento de tráfico en la red de manera convencional. Este último es el tipo de firewall que estudiaremos en el presente documento.

7.1.13.2 Los Firewalls y su relación con las Políticas de Seguridad

Un firewall divide el mundo en dos áreas discretas: el mundo exterior y el mundo interno de la organización, cada una de las cuales es atendida por una (o varias) interfaces de red.

Adicionalmente, el administrador define un conjunto de reglas de tráfico para controlar el acceso de un área de la red corporativa a otra e incluso, hacia afuera de la misma. Tales reglas dependen de lo que se conoce comúnmente como una Política de Seguridad.

Una adecuada política de seguridad se denomina a la colección de decisiones que una organización toma acerca de la seguridad en una red y la reglamentación acerca de que actividades se permiten o cuales se prohíben. 1

---

1 Tomado de SunScreen EFS 3.1. **Reference Manual**. "Security Considerations"

Por este motivo, el aspecto más importante en la instalación y administración de un firewall, es la creación de una política de seguridad bien definida.

Cuando se hace la definición de una política de seguridad, deben considerarse los siguientes aspectos:

- Qué servicios necesitan acceder los empleados de su empresa?
- A qué servicios o información necesitan tener acceso los clientes u otros usuarios de Internet?
- Contra qué o contra quiénes intenta usted proteger a su compañía?

Una política de seguridad es una directiva de protección y por lo tanto es necesario determinar los activos que se intenta proteger y saber contra que o contra quienes. Una vez que se han identificado los requerimientos de seguridad para la protección de la integridad así como la accesibilidad a la información de la compañía y a los recursos computacionales, debe determinarse que servicios se quieren soportar en el sitio para los empleados y para los clientes.

Para ayudar a determinar tales requerimientos, se sugiere analizar y responder las siguientes preguntas:

- Necesitan los usuarios transferir archivos fuera de la organización?
- Los usuarios descargarán archivos desde fuera de la organización?
- Qué tipo de acceso a la información de la compañía se le brinda a los clientes?

- Quiénes necesitan usar servicios locales desde sitios remotos?
- Es necesario usar direcciones privadas para poder soportar otros firewalls o subredes de las que están disponibles de acuerdo a los recursos asignados por el ISP (Internet Service Provider)?
- Se necesitan direcciones privadas para re-enumerar más fácilmente la red, los firewalls y los hosts?
- Es necesario usar direcciones de Internet sin registrar?

Una vez que se hayan determinado las respuestas a estos y otros problemas de seguridad específicos, se tendrá una base para hacer una configuración de la red que utilizará como un medio de protección un firewall.

Las reglas y políticas de seguridad en el firewall, son utilizadas para controlar el acceso a los datos. Por defecto, los firewalls rechazan los paquetes que no cumplan con una regla o política definida. Esto hace mucho más fácil la creación de reglas para la política de filtro de paquetes, puesto que únicamente se definen reglas para los servicios a los que se les permite el acceso el acceso a la red.

La implementación de las políticas de red, tiene tres etapas definidas a saber: 1

- Identificación de los servicios que estarán disponibles en la red.

---

1 Ibid. "Security Considerations"

- Identificación de los servicios disponibles para un usuario particular o para un grupo de usuarios y sus direcciones IP's.
- Establecimiento mediante políticas en el firewall de las acciones concretas a tomar para los servicios y las direcciones de cada usuario.

7.1.13.3 Los Firewalls de tipo Stealth

Los firewalls configurables en modo stealth, son poco conocidos, pero a la vez constituyen un paradigma interesante de las arquitecturas de firewall. El primer paso en un ataque informático es "Conocer al Enemigo" y por tal motivo, un atacante en primera instancia prueba, revisa y enumera todos los recursos visibles de la Intranet. Sin embargo, los firewalls stealth son difíciles de identificar en la red por lo que dificultan al atacante la labor de conocimiento de la topología de red existente dentro de la organización.

Debido a que no utilizan direcciones IP de ningún tipo (públicas registradas o sin registrar, privadas, especiales, etc.), los firewalls en modo stealth proveen un mayor nivel de seguridad contra ataques cuando las organizaciones se conectan a redes publicas inseguras. Por este motivo, las interfaces de red de un firewall en modo stealth no

tienen direcciones IP ni tampoco hacen el enrutamiento entre redes conocido también como "Bridging" 1

El modo Stealth, generalmente se instala haciendo un aseguramiento del sistema operativo o "hardening"2 del sistema operativo mediante la eliminación de los paquetes y componentes que no son utilizados por el software de firewall. Esto permite que, si por ejemplo se detecta un bug de seguridad en el servicio de mail de un servidor Sun Solaris, se pueda proteger al servidor de tal bug al remover todos los paquetes del sistema operativo que tengan como propósito habilitar el servicio de correo.

Adicionalmente, el firewall Stealth no requiere hacer particionamiento de la red debido a que actúa examinando una existente y subsecuentemente no necesita segmentar aún más la red como en el caso de los firewalls instalados en modo de enrutamiento.

Los firewalls en modo Stealth son típicamente usados para defensa del perímetro dedicada y extranets y no son accesibles vía una dirección IP.

Se tienen en cuenta 2 requisitos fundamentales para la consideración de un firewall como verdaderamente de tipo stealth. Primero, un firewall

---

1. El enrutamiento entre redes también denominado bridging es suprimido de los firewalls en modo Stealth cuando manipulan paquetes al nivel de la capa MAC con el fin de evitar su detección debido a la disminución del TTL (Time To Live)

2. Se conoce como "Hardening" al proceso mediante el cuál se deshabilita la mayor cantidad de servicios posibles en una estación o servidor de la red con el fin de minimizar la vulnerabilidad del mismo y mejorar su rendimiento.

stealth no tiene direcciones o presencia en el nivel tres (3) del modelo OSI en una red para la cual se está prestando el servicio de control de acceso y filtro de paquetes. A cambio de esto, el firewall se comporta como un switch conectado a múltiples segmentos de red al mismo tiempo que provee mecanismos de control de acceso en ese punto. El segundo requerimiento es una implicación del primero: los firewalls stealth no decrementan el TTL (Time To Live) en los paquetes que manejan.1 Por este motivo, cualquiera que esté vigilando en una red la ruta que siguen los paquetes, no podrá detectar el firewall stealth como un nodo más de la red (también conocido como hop).

Solamente algunos productos del mercado tienen la capacidad actualmente de funcionar como firewalls en modo stealth. Se puede mencionar como ejemplo, aquellos incorporados con los sistemas operativos FreeBSD, Linux y el SunScreen Secure Net 3.1 de Sun Microsystems. Los dos primeros, combinan el filtrado de paquetes con un modo especial de enrutamiento en modo bridge2 para simular el modo stealth mediante el uso del cornando ipfilter.

SunScreen Secure Net, en cambio, lo hace a través de una versión especial de una máquina de estados que puede ser habilitada al

---

1. GILLESPIE, Brandon. **"Stealth Firewalls".** Sans Institute. Information Security Reading Room. Abril 10 de 2001.

2. Como se mencionó anteriormente, el modo bridge permite que los paquetes que son enrutados de una red a otra, no sean analizados por el stack de IP del sistema operativo, evitando así, la reducción de su TTL y por consiguiente su detección en la red.

momento de la instalación del producto. Según los analistas1, SunScreen Secure Net se muestra como el más maduro de estos productos. Sin embargo, a la hora de considerar costos, FreeBSD y Linux están disponibles en forma gratuita, mientras que SunScreen Secure Net cuesta alrededor de U$ 4500. Este valor no incluye los costos administrativos ya que si bien FreeBSD y Linux requieren de un conocimiento especializado y una administración compleja, SunScreen Secure Net es un producto completo y altamente compacto. Actualmente, varias compañías están desarrollando software que permite acoplarse en forma rápida y precisa con SunScreen como es el caso de WebTrends (www.webtrends.com) que fabrica un producto diseñado para llevar a los administradores de redes e investigadores forenses, un método mejorado para el análisis de logs y detección de tráfico sospechoso mediante herramientas que facilitan el examen estadístico de los servicios que se ofrecen en la red.

7.1.13.4 La Arquitectura Stealth

Los firewalls stealth pueden manejar varias interfaces de red, sin embargo no son enrutadores. Estos últimos dividen una subred en distintos segmentos o zonas con el firewall en el centro, acoplándolas todas en forma similar a como lo haría un hub o un switch. Los firewalls stealth hacen esto a un nivel de red inferior al que se utiliza para hacer

1. GILLESPIE, Brandon. **"Stealth Firewalls".** Sans Institute. Information Security Reading Room. Abril 10 de 2001.

el enrutamiento y en cambio examinan cada paquete IP en forma similar a la que lo hacen los sniffers de red, moviéndose entre interfaces según sea apropiado y basándose en listas de control de acceso (ACL: Access Control List). Para lograr esto, el firewall debe saber cuales hosts están conectándose a través de que interfaz lo cual puede ser visto como una posible manera de detección de un firewall stealth porque al hacer el respectivo análisis comparativo, un hacker podría percatarse del acceso permitido desde ciertos hosts pero no desde otros. Adicionalmente, las listas de acceso, deben configurarse de manera que puedan describir qué tráfico está permitido desde y hacia los diferentes segmentos.

7.1.13.5 Ventajas de los Firewalls tipo Stealth

En comparación con los firewalls que hacen enrutamiento, los firewalls en modo stealth tienen algunos méritos de ventaja debido a que, entre otras causas, todos los ataques son limitados y se restringen al nivel dos (2) del modelo OSI, puesto que el firewall no tiene una clara presencia en el nivel tres (3) del modelo en la red que está protegiendo.

De este modo, es más complicado determinar que producto y que versión de firewall se está usando. Así, se dificulta enumerar en forma externa los nodos de la Intranet y determinar la topología de una red existente. Adicionalmente, las redes actuales pueden ser protegidas por un firewall sin tener que ser divididas en subredes.

Una de las formas de derrotar a un firewall es atacarlo directamente. Usualmente, este es un dispositivo de red como cualquier otro y tiene una dirección IP. Cualquier individuo puede ocasionalmente probar y lanzar requerimientos contra esta dirección con el fin de obtener información útil como el tipo de producto de firewall que se está utilizando, el nivel de patches instalados e incluso identificar que servicios se están ejecutando en la máquina. Cuando el firewall trabaja en modo Stealth, esta labor se hace complicada.

Al no tener presencia en el nivel tres (3) del modelo OSI en la red, no es posible atacar y/o probar el firewall. Esto limita el número de ataques directos contra el firewall a únicamente los que son realizados sobre el nivel 2. Además de ser un blanco fácil, esto también significa que es difícil determinar que tipo de producto de firewall se está utilizando y que versión del mismo se está ejecutando. Adicionalmente, se puede ubicar un firewall stealth en una Intranet, uniendo segmentos de red separados y estableciendo relaciones entre los segmentos aún dentro de la misma subred. De este modo es mucho más difícil determinar la topología de red porque externamente no es evidente determinar en que punto de la red interna está ubicado el firewall en relación con los demás hosts. Con un firewall en modo enrutamiento, se podría determinar fácilmente esto mediante la deducción de qué subredes está protegiendo.

La habilidad para proteger de ataques a una red sin necesidad de segmentarla puede ser muy útil durante la adición de seguridad a una

red existente, puesto que no es necesario cambiar la topología de las subredes que actualmente se tiene implementada.

Pero, en qué punto se encuentra el secreto de la tecnología Stealth? Podemos afirmar que se encuentra concentrado básicamente en el manejo que se hace a nivel interno de los paquetes de red.

Aunque normalmente se efectúa la retransmisión de paquetes de datos (forwarding) y bloqueo de otros (packet filtering), un firewall típico se comporta como lo haría un enrutador: decrementando el TTL en los paquetes procesados y confirmando a toda la red que efectivamente, en este punto existe un nodo. Pero en realidad sí es posible ocultar la presencia de un firewall de aplicaciones inquisitivas como el traceroute de Unix que utiliza paquetes UDP con varios valores de TTL para ubicar e identificar los nodos existentes entre dos hosts de la red.

Si al trabajar en sistemas Linux y FreeBSD (se puede generalizar a cualquier sistema Unix) se desea que los paquetes de traceroute funcionen pero no se quiere anunciar la presencia del firewall como un nodo más de la red, se debe definir una regla a nivel de manejo del tráfico como la siguiente:

block in quick on hme0 fastroute proto udp from any to any port 33434 >< 33465

La presencia del comando clave fastroute, dará la señal al software de manejo de paquetes ipfilter de no permitir el paso de los paquetes dentro del stack IP de Unix para que sean enrutados, lo que resultaría en un decremento del TTL.

Por el contrario, el paquete será puesto delicadamente en la interfaz de salida por el sistema de filtrado de paquetes y no sucederá tal decremento.

El sistema de filtrado de paquetes usará desde luego la tabla de enrutamiento del sistema para decidir cuál es realmente la interfaz de salida apropiada, pero tendrá el cuidado apropiado con la tarea misma de enrutamiento.

Adicionalmente, la presencia del comando block in quick, se utiliza para evitar que al usar la retransmisión manual de paquetes con el comando pass y al tener el sistema de IP forwarding habilitado en el kernel del sistema operativo1, este tenga dos caminos por donde enviar el paquete y así probablemente el kernel se iría a modo pánico.

Este es básicamente el secreto de los firewalls tipo Stealth. Un concepto sencillo pero interesante y con múltiples posibilidades.

---

1. En general, todos los sistemas operativos modernos basados en Unix como Linux, Solaris, FreeBSD, Tru64, AIX y HP-UX entre otros, poseen características para el tratamiento de paquetes de red como el *IP forwarding* que determina si un host realizará o no labores de enrutamiento de paquetes IP.

7.1.13.6 Conclusiones

Un administrador de redes debe considerar cuidadosamente cada uno de los aspectos que involucran el diseño de una red que van desde la planeación de la topología más adecuada hasta la escogencia de los mecanismos de defensa contra ataques que vulneran la seguridad de los sistemas internos.

Tales mecanismos deben ser adoptados teniendo en cuenta elementos tan importantes como el acceso de los usuarios a la red, los aplicativos utilizados y los servicios que estarán disponibles tanto para los usuarios internos como los externos.

De este modo, se hace necesaria la utilización de tecnologías como los firewalls Stealth que si bien protegen la red de igual manera que los firewalls en modo enrutamiento, tienen ciertas ventajas frente a estos tales como:

- Dificultad para ser detectados en la red con métodos convencionales
- Restringen el envío de información que revele características de la red
- Evitan la repetición del proceso de segmentación de las redes para poder utilizar una o varias direcciones IP
- Permite analizar el tráfico de paquetes de la red para detectar tráfico indeseado y detenerlo

- Dificulta los ataques informáticos basados en niveles del modelo OSI superiores al nivel de red.

Dicha tecnología de firewalls stealth se hace posible gracias una idea que maneja en forma inteligente el tráfico de paquetes que van desde una interfaz de red del servidor conectada en un segmento de la misma, hasta otra interfaz de red en el mismo servidor que hace las veces de firewall stealth.

Los firewalls en modo stealth, manejan políticas de enrutamiento de paquetes basadas en ACL's (Access Control Lists) que les permiten transportar los paquetes en la red sin disminuir su TTL haciendo indetectable su presencia en la red (nivel 3) mediante métodos tradicionales.

Finalmente, es importante mencionar que el tipo de firewall que se adopte en la red no solamente debe instalarse con el fin de impedir los ataques a la red interna, sino que también debe estar estrechamente ligado a las políticas de crecimiento de la red y de acceso a los recursos para los usuarios. Es a la vez fundamental escoger un producto que se adecúe a las necesidades de la organización y que permita niveles de confiabilidad acordes con dichas necesidades. De esta manera, productos en el área de la seguridad en redes como SunScreen Secure Net, pueden ser una opción para empresas con necesidades críticas de seguridad en su infraestructura de red. Si por el contrario, el tema no es tan dramático debido entre otras cosas a que el tipo de información que se

transmite a través de la red no supera los umbrales de confidencialidad mínimos, productos como el firewall de FreeBSD o de Linux serán suficientes para adoptar un esquema de seguridad.

Cualquiera que sea la escogencia del producto a instalar, lo realmente importante es que el lector sea consciente de que en las redes públicas, el aseguramiento es un proceso permanente y de afinamiento constante.

7.1.14 Factores que no hacen deseable un Firewall

7.1.14.1 Ineficiente

El Firewall se convierte en un cuello de botella de toda la estructura y debe poseer por lo tanto una eficiencia en la manipulación de los streams de paquetes que sea igual o superior a la del enrutador que maneja tal enlace.

Normalmente la experiencia y conocimiento de los fabricantes de Firewalls no se acerca siquiera a la tradición y conocimiento de los fabricantes tradicionales de enrutadores, por ello rara vez pueden cumplir el requisito anterior y lo que se consigue en la practica es un cuello de botella, así como enrutadores sub utilizados debido a la situación anterior.

Este factor también nos conduce a que los costos para máquina de Firewall que cumplan tales requisitos sean bastante altos ya que su

volumen de producción (numero de unidades vendidas) no se acerque a la producción típica de los enrutadores correspondientes para ese nivel de procesamiento de paquetes por segundo.

7.1.14.2 No tan seguro

Los Firewalls son típicamente implementados en un sistema UNIX lo que los hace bastante vulnerables para los ataques de seguridad, ya que de tal sistema existe mayor conocimiento del público en general, y son bastante publicadas las posibles brechas de seguridad en ese sistema operativo, por ello es el blanco típico de ataque para los programas especializados de scanning de los. Estos programas son en un 99% desarrollados para sistemas UNIX.

Si mi seguridad está sustentada en una máquina cuyo núcleo está apoyada en el sistema UNIX (el cual es precisamente el más conocido por los enemigos de mi seguridad), entonces mi sistema no es realmente tan seguro.

7.1.14.3 No son transparentes a la operación del usuario

Debido a su diseño, algunos de estos modelos no son tan transparentes a la operación del sistema, complican la administración del sistema de comunicación (usualmente tienen interfaces de manejo propietarias).

Algunos modelos basados en "proxies" pueden ser muy seguros, pero algunos de ellos requieren versiones modificadas de los aplicativos, llevándolos a ser poco deseables para montajes masivos.

7.1.14.4 Inapropiados para implementación mixta

Por su misma concepción la implementación solicitada por las compañías cuenta con dos niveles de VPNs (la intranet corporativa y luego las intranet de cada empresa), los cuales deben ser interrrelacionados de manera armoniosa para flujo de información y control de acceso.

Este tipo de implementación seria bastante costoso, difícil de implementar y de administrar con dos niveles de Firewalls.

7.1.15 Comprar o Construir

Algunas organizaciones tienen la capacidad de colocar sus propios Firewalls, usando cualquiera de los equipos y componentes de software disponibles o escribiendo una Firewall. Al mismo tiempo, la totalidad de los vendedores ofrecen una amplia variedad de servicios en tecnología de Firewall, desde proveer las herramientas necesarias hasta implementar pólizas de seguridad hasta cálculos fuera de riesgos, revistas de seguridad y entrenamiento de seguridad.

Una de las ventajas para una compañía al construir su propio Firewall es que el personal de la misma entenderá las especificaciones del diseño y uso del mismo. Tal conocimiento puede no existir para el vendedor - proveedor del Firewall. Además, un Firewall puede requerir una gran cantidad de tiempo para construirla, documentarla y mantenerla.

Un Firewall puede ser tan efectiva como la administración que la hizo. Un mantenimiento pobre puede empezar a ser inseguro y permitir roturas mientras provee una ilusión de seguridad. La póliza de seguridad podría reflejar claramente la importancia de la administración de una Firewall fuerte, y el manejo demostraría su importancia en términos de personal, fondos y otros recursos necesarios.

El contar con un Firewall no es excusa para prestar menos atención a la administración de un sistema en el lugar, de hecho, si un Firewall es penetrado, una administración pobre permitirá amplias intrusiones resultando dañada, además no reduce las necesidades de administrar un sistema altamente calificado al mismo tiempo.

Amoroso y sharp concuerdan en que no es sencillo y correcto el set de funciones de un Firewall para todos los medio ambientes.

Ellos recomiendan que cada comprador seleccione funciones basadas en los requerimientos únicos de la empresa que desee contar con un Firewall.

Un problema encontrado por muchos compradores de Firewall es que los vendedores, preparan literatura que ponen a sus productos en lo más alto posible y describen diseños y filosofías de ventas apropiadas para la compañía. Sin embargo. Los estándares han surgido en otras áreas de hardware y software, ambos en tecnología y descripción de funciones.

7.1.16 Certificación

ICSA, Inc. Intenta desarrollar criterios imparciales para definir buenos productos de seguridad. Por ejemplo, por muchos años ICSA ha estado probando y certificando productos antivirus. Los usuarios de estos productos han indicado que la certificación ha sido de gran ayuda. Una compañía compra un producto antivirus certificada sabe que realizara estándares claros establecidos y de esta manera podrá evitar más desordenes costosos que de otra manera requerirá de otra clase de diligencias.

La certificación Firewall opera con principios similares. Las compañías que fabrican Firewall pueden someterlos a prueba, y si pasan la prueba, ellos pueden colocar el logo de certificación. Esto proporciona una seguridad a los compradores que este producto satisface ampliamente un nivel de estándar de seguridad. En otras palabras, un comprador puede confiar que todos los productos que han sido certificados, realizan, en una perspectiva de seguridad, funciones en un mismo nivel.

Por supuesto, algunos productos exceden el nivel y en algunas áreas la certificación será más y más severa (la certificación puede ser revocada si un producto falla al no mantenerse con el estándar).

La certificación ICSA es totalmente diferente de un análisis competitivo o examen de producto. El propósito de la certificación es no decir que un producto "A" hace o realiza mejor que el producto "B" respecto a esto o aquello. Es únicamente la ejecución relativa de pruebas lo que cuenta, como un paso binario/resultados fallidos. La realización de un producto en términos de velocidad, no es parte de la certificación.

El estándar inicial de certificación Firewall depende de una definición de requerimientos mínimos aceptables para una compañía típica o una organización. Específicamente, los criterios de certificación significan que un producto Firewall, que ha sido configurado de acuerdo a las instrucciones del fabricante, brinda protección contra ataques y al mismo tiempo, brinda una organización con funcionalidad operativa real.

La certificación está diseñada para asegurar que un Firewall repela importantes ataques, comunes y no comunes. ICSA utiliza una variedad de herramientas de rastreos comerciales e internos así como técnicas manuales para verificar que los ataques son combatidos efectivamente. Esto asegura que hay una fundación confiable de buenas técnicas de seguridad. El Firewall debe proveer una organización con una funcionalidad real. Los usuarios pueden accesar al internet, pueden

conectarce a sistemas internos a través de la Firewall, puede una organización enviar y recibir correos a través del Firewall, etc.

Los Firewall certificadas por ICSA no garantizan que sean impenetrables. Un buen producto podría ser instalado inapropiadamente, permitiendo vulnerabilidades.

7.1.17 Procesos de prueba

- La guía de prueba es suministrada por el vendedor y será revisada por el staff del laboratorio ICSA para su total exactitud.
- El sistema operativo será instalado en una maquina "limpia" y almacenado de acuerdo a las instrucciones del vendedor.
- El Firewall será instalada en ausencia de configuración de instalación, en cada servicio abierto se revisará que este apoyado apropiadamente.
- El Firewall será rastreada en ausencia de configuración.
- El Firewall será configurada para apoyar el perfil de servicios requeridos por ICSA, y cada servicio abierto se chocara que este apoyado apropiadamente.
- El Firewall será rastreada mientras apoya el perfil de servicios requerido por ICSA.
- Las funciones de entrada y salida serán revisadas de conformidad.

CAPÍTULO 8

8. IMPLEMENTACIÓN DE SISTEMAS DE DETECCIÓN DE INSTRUSOS

A pesar de que un enfoque clásico de la seguridad de un sistema informático siempre define como principal defensa del mismo sus controles de acceso (desde una política implantada en un Firewall hasta unas listas de control de acceso en un router o en el propio sistema de ficheros de una máquina), esta visión es extremadamente simplista si no tenemos en cuenta que en muchos casos esos controles no pueden protegernos ante un ataque. Por poner un ejemplo sencillo, pensemos en un Firewall donde hemos implantado una política que deje acceder al puerto 80 de nuestros servidores Web desde cualquier máquina de Internet; ese Firewall sólo comprobará si el puerto destino de una trama es el que hemos decidido para el servicio HTTP, pero seguramente no tendrá en cuenta si ese tráfico representa o no un ataque o una violación de nuestra política de seguridad: por ejemplo, no detendrá a un pirata que trate de acceder al archivo de contraseñas de una máquina aprovechando un bug del servidor Web. Desde un pirata informático externo a nuestra organización a un usuario autorizado que intenta obtener privilegios que no le corresponden en un sistema, nuestro entorno de trabajo no va a estar nunca a salvo de intrusiones.

Llamaremos intrusión al grupo de acciones que intentan comprometer la

integridad, confidencialidad o disponibilidad de un recurso; analizando esta definición, podemos darnos cuenta de que una intrusión no tiene por qué consistir en un acceso no autorizado a una máquina: también puede ser una negación de servicio. A los sistemas utilizados para detectar las intrusiones o los intentos de intrusión se les denomina sistemas de detección de intrusiones (Intrusion Detection Systems, IDS), y aunque no sea la traducción literal - sistemas de detección de intrusos; cualquier mecanismo de seguridad con este propósito puede ser considerado un IDS, pero generalmente sólo se aplica esta denominación a los sistemas automáticos (software o hardware): es decir, aunque un guardia de seguridad que vigila en la puerta de la sala de operaciones pueda considerarse en principio como un sistema de detección de intrusos, como veremos a continuación lo habitual (y lógico) es que a la hora de hablar de IDS no se contemplen estos casos.

Una de las primeras cosas que deberíamos plantearnos a la hora de hablar de IDS es si realmente necesitamos uno de ellos en nuestro entorno de trabajo; a fin de cuentas, debemos tener ya un sistema de protección perimetral basado en Firewall, y por si nuestro Firewall fallara, cada sistema habría de estar configurado de una manera correcta, de forma que incluso sin Firewall cualquier máquina pudiera seguirse considerando relativamente segura. La respuesta es, sin duda, sí; debemos esperar que en cualquier momento alguien consiga romper la seguridad de nuestro entorno informático, y por tanto hemos de ser capaces de detectar ese problema tan pronto como sea posible (incluso antes de que se produzca,

cuando el potencial atacante se limite a probar suerte contra nuestras máquinas). Ningún sistema informático puede considerarse completamente seguro, pero incluso aunque nadie consiga violar nuestras políticas de seguridad, los sistemas de detección de intrusos se encargarán de mostrarnos todos los intentos de multitud de piratas para penetrar en nuestro entorno, no dejándonos caer en ninguna falsa sensación de seguridad: si somos conscientes de que a diario hay gente que trata de romper nuestros sistemas, no caeremos en la tentación de pensar que nuestras máquinas están seguras porque nadie sabe de su existencia o porque no son interesantes para un pirata.

8.1 La necesidad de un IDS

Dentro de las soluciones tecnológicas que en la actualidad están disponibles para reforzar la seguridad de una red, los firewalls son muy populares.

Un firewall es un sistema encargado del cumplimiento de las políticas de control de acceso a la red, lo cual se hace a través de reglas. Un firewall actúa como guardia perimetral de una red: protege una red de ataques que provengan del exterior de ésta. Pero el escenario se puede complicar de la siguiente forma:

- Un atacante puede lograr pasar el firewall, dejando la red a su merced.

- Un firewall protege de los accesos no autorizados hacia la red interna, pero no protege a las máquinas ubicadas en la red perimetral como servidores Web, servidores de correo, servidores FTP, en otras palabras, a las bases funcionales de Internet.
- Un firewall no protege contra ataques desde adentro.

En estos casos lo que nos queda es detectar el ataque o la intrusión lo antes posible para que cause el menor daño en el sistema. Antes de continuar vamos a definir qué se entiende normalmente por intrusión. Normalmente un intruso intenta:

- Acceder a una determinada información.
- Manipular cierta información.
- Hacer que el sistema se no funcione de forma segura o inutilizarlo.

Una intrusión es cualquier conjunto de acciones que puede comprometer la integridad, confidencialidad o disponibilidad de una información o un recurso informático. Los intrusos pueden utilizar debilidades y brechas en la arquitectura de los sistemas y el conocimiento interno del sistema operativo para superar el proceso normal de autenticación.

La detección de intrusos se puede encontrar a partir de la caracterización anómala del comportamiento y del uso que hacen de los recursos del sistema. Este tipo de detección pretende cuantificar el comportamiento normal de un usuario. Para una correcta distinción hay que tener en

cuenta las tres distintas posibilidades que existen en un ataque, atendiendo a quién es el que lo lleva a cabo:

- **Penetración externa:** Que se define como la intrusión que se lleva a cabo a partir un usuario o un sistema de computadores no autorizado desde otra red.
- **Penetraciones internas:** Son aquellas que llevan a cabo por usuarios internos que no están autorizados al acceso.
- **Abuso de recursos:** Se define como el abuso que un usuario lleva a cabo sobre unos datos o recursos de un sistema al que está autorizado su acceso.

La idea central de este tipo de detección es el hecho de que la actividad intrusiva es un subconjunto de las actividades anómalas. Esto puede parecer razonable por el hecho de que si alguien consigue entrar de forma ilegal en el sistema, no actuará como un usuario normal. Sin embargo en la mayoría de las ocasiones una actividad intrusiva resulta del agregado de otras actividades individuales que por sí solas no constituyen un comportamiento intrusivo de ningún tipo. Idealmente el conjunto de actividades anómalas es el mismo del conjunto de actividades intrusivas, de todas formas esto no siempre es así:

- **Intrusivas pero no anómalas**: Se les denomina falsos negativos y en este caso la actividad es Intrusiva pero como no es anómala y

no se consigue detectarla. Se denominan falsos negativos porque el sistema erróneamente indica ausencia de intrusión.

- **No Intrusivas pero anómalas:** Se denominan falsos positivos y en este caso la actividad es no Intrusiva, pero como es anómala el sistema decide que es Intrusiva. Se denominan falsos positivos, porque el sistema erróneamente indica la existencia de intrusión.
- **Ni Intrusiva ni anómala:** Son negativos verdaderos, la actividad es no Intrusiva y se indica como tal.
- **Intrusiva y anómala:** Se denominan positivos verdaderos, la actividad es Intrusiva y es detectada.

Los primeros no son deseables, porque dan una falsa sensación de seguridad del sistema y el intruso en este caso puede operar libremente en el sistema.

Los falsos positivos se deben de minimizar, en caso contrario lo que puede pasar es que se ignoren los avisos del sistema de seguridad, incluso cuando sean acertados. Los detectores de intrusiones anómalas requieren mucho gasto computacional, porque se siguen normalmente varias métricas para determinar cuánto se aleja el usuario de lo que se considera comportamiento normal.

En caso de que exista la suficiente certeza de la detección de un incidente, el IDS tiene como función principal alertar al administrador o personal de seguridad, para que tome acciones al respecto.

Otras implementaciones más complejas son capaces de ir más allá de la notificación de un posible ataque, es decir pueden ejecutar acciones automáticas que impidan el desarrollo de éste.

Es necesario implementar una serie de chequeos de seguridad a diferentes capas dentro de los componentes de TI.

8.2 Qué es un Sistema de Detección de Intrusos?

Un IDS (Sistema de Detección de Intrusiones) detecta y alerta sobre las intrusiones intentadas en un sistema o en una red, cuando se considera que una intrusión es toda actividad no autorizada o no deseada ocurrida en ese sistema o red.

¿Pero, eso no es lo que hace una barrera "corta fuego" (firewall)?

No. Los sistemas de detección de intrusiones son una combinación de los sistemas que alertan anticipadamente y los que alarman cuando algo ha ocurrido, y esto es muy diferente a lo que realizan los Firewalls.

El rol primario de una barrera corta fuego es limitar el acceso entre redes. Las barreras corta fuego están diseñadas para filtrar el tráfico "normal" de la red, basándose en atributos tales como las direcciones de origen y de destino, números de puerto, etc. Si bien las barreras de fuego más modernas también se preocupan por los requisitos de los protocolos populares, como DNS, con frecuencia no manejan correctamente el tráfico de red "incorrecto", catalogado de malicioso.

En comparación, un IDS que analiza la Red ("NIDS") se ocupa de lo que constituyen paquetes de la red legales y de los ilegales y puede generar alertas cuando se detectan estos últimos. Sin embargo, generalmente, un IDS no previene ni contesta un ataque.

Se puede dividir la infraestructura de TI (Tecnología de Información) en 3 grandes partes y a su vez, mencionar el tipo de Seguridad que es necesario aplicar a cada una de ellas:

- Servidores y Clientes – Seguridad Perimetral
- Seguridad a nivel Servidor
- Seguridad a nivel Cliente

8.2.1 Seguridad Perimetral

Dentro de la Seguridad Perimetral se encuentran los dispositivos como Firewalls y Routers, servidores de VPN, dispositivos de Antivirus para correo entrante y saliente de la red, mecanismos para filtrado de contenido etc. De estos, el firewall es la primera línea de defensa en una red, y como tal, no debe de ser considerado como la "única" solución. De hecho, no existe ningún producto único o solución única, que pueda resolver el problema de seguridad a todos los niveles en la infraestructura de TI.

8.2.2 Seguridad a Nivel Servidor

Siguiendo en el esquema de seguridad, llegamos a la Seguridad a nivel Servidor. Aquí es donde se protegen y monitorean aquellos sistemas que son críticos para el funcionamiento y la misión de la organización, y cuya confiabilidad, disponibilidad y privacidad es simplemente vital.

8.2.3 Seguridad a Nivel Cliente

En esta parte es donde es de primordial importancia tener soluciones que salvaguarden a todos los sistemas que son parte de la red, ya sean desktops, laptops etc. La seguridad de una fortaleza es tan fuerte como su puerta mas débil lo es. Esto quiere decir que al desarrollar el esquema de seguridad a profundidad, debemos de enfocarnos también de aquellas maquinas que parecerían no tener tanta relevancia dentro del esquema de red, pero que sin embargo si tienen todos los accesos necesarios a los servidores de mayor importancia.

Algunas soluciones de seguridad en esta rama son tales como el Symantec Client Security, que unifica en una sola solución las funcionalidades de Antivirus, Firewall, Detector de Intrusos y Filtrado de Acceso a nivel de cada cliente, y que puede ser administrado desde un lugar central.

En la Seguridad a nivel de servidor es donde se implementan soluciones como Intruder Alert, que es un sistema de Detección de Intrusos a nivel de host. Esto quiere decir que Intruder Alert puede monitorear la hora y el momento de acceso remoto o local, modificación de cuentas,

modificación y acceso de archivos, cambios en el registro, y muchos otros detalles de aquellos servidores bajo su cuidado.

Un sistema de detección de Intrusos como Intruder Alert puede informar, documentar, alertar y reaccionar frente a acciones malignas, como el robo y modificación de información confidencial. Intruder Alert no solamente protegerá a sus sistemas de ataques realizados por usuarios externos a su red, sino también de aquellos usuarios locales que tengan algún tipo de acceso a dichos servidores.

8.3 Características de un IDS

Es necesario enfocar algunas características que un buen sistema debe de tener. Estas características son:

1. Debe de ejecutar continua y autónomamente, sin necesidad constante de supervisión humana. Los procesos deberán de ser no-intrusos con respecto a las aplicaciones y comunicaciones que se desarrollan dentro de los servidores. El sistema deberá de ser suficientemente confiable para poder ejecutar en el background de los sistemas monitoreados.
2. Debe de ser capaz de soportar fallas en el sistema que monitorea, poder sobrevivir y reiniciar ejecución en caso de que el sistema en el que reside se caiga y tenga que ser reiniciado.
3. Debe de poder resistir subversión, es decir que el sistema de Detección de Intrusos debe de ser capaz de monitorearse a él mismo.

4. Deberá de no ser una carga para el sistema monitoreado. Deberá de haber, si acaso, no mas que un pequeño efecto notable cuando se ejecute el sistema de Detección de Intrusos.
5. Deberá de ser capaz de observar variaciones a aquello que se considera comportamiento normal.

Los sistemas de detección de intrusos proporcionan tres funciones esenciales de seguridad: monitorean, detectan y responden a la actividad que consideran sospechosa. Veamos un poco más en detalle qué significa cada acción.

Monitorean: El IDS mantiene siempre un ojo en la red, observando y escudriñando el tráfico en busca de cualquier paquete susceptible de contener código no deseado. Qué visita quién y cuándo lo hace en nuestra red, quién viene desde el exterior y qué busca... etc.

En este sentido actúa exactamente igual que un sniffer. De hecho, cabe la posibilidad de utilizarlos como tal. Para llevar a cabo esta tarea, debe interceptar todos los paquetes de información y consultar los campos de destino, origen, puertos, etc.

Detectan: Usan políticas (reglas totalmente configurables) para definir los actos sospechosos de todo ese tráfico que provocarán una alarma si coinciden. Las reglas deben ser actualizadas cada poco tiempo, pues el tráfico sospechoso varía según se descubren nuevas vulnerabilidades.

Responden: Esta alarma puede venir en forma de ejecución de archivos en el sistema, páginas html dinámicas con gráficos o incluso correos con la información necesaria. También podría incluir la expulsión de un usuario del sistema, el aislamiento de la máquina mediante la desconexión de la tarjeta de red... etc.

Normalmente, se instala un sistema Web protegido con contraseña, donde el administrador puede consultar desde cualquier lugar con conexión a Internet qué está ocurriendo en tiempo real en su sistema.

Direcciones y puertos

Nuestro servidor Web para Internet, nuestro servidor seguro para comercio electrónico y nuestro servidor de correo electrónico usan números de puerto diferentes, a los cuales se accede desde la dirección Internet de la empresa. Este direccionamiento numérico tan poco amistoso normalmente está asociado con un nombre de dominio más aceptable, del tipo www.nuestraempresa.com.ec

Las barreras corta fuego están diseñadas para filtrar el tráfico de la red sobre la base de direcciones IP y números de puerto que uno desea hacer accesibles.

De manera que nuestras barreras corta fuego podrían solamente permitir el tráfico destinado a nuestro servidor de correo (número de puerto 25) o a nuestro servidor de Web (número de puerto 80).

Hemos sido hackeados porque nuestras barreras de fuego no miran lo que realmente se envía a esos puertos. Quizás el mensaje no era un pedido legítimo de una página Web, sino algo mucho más siniestro. Para inspeccionar el contenido de esos mensajes, necesitamos un IDS.

Una barrera corta fuego no puede decirnos cuáles archivos del sistema habían estado en nuestros servidores. Para eso, se requiere un host basado en IDS.

Finalmente, si alguien ha conectado un módem a uno de nuestros sistemas, el tráfico de Internet estará ignorando totalmente a nuestra barrera corta fuego pero no al IDS

8.4 Mecanismos de detección del mal uso o uso sospechoso

Los sistemas de detección de intrusos se pueden caracterizar entre aquellos que se basan en:

- La detección del mal uso.
- La detección del uso anómalo.

La detección del mal uso en un IDS involucra la verificación sobre tipos ilegales de tráfico de red; por ejemplo, combinaciones de opciones dentro de un paquete que nunca podrían ocurrir legítimamente. La detección del

mal uso en este tipo de IDS podría incluir a los intentos de un usuario por ejecutar programas, cuando no tiene una necesidad legítima de hacerlo.

La detección de actividad anómala se apoya en que el sistema conoce cual es el tráfico "regular" en la red y por ende el que no lo es. Un ejemplo de tráfico anómalo en un NIDS es el acceso que se intenta repetidamente desde una máquina remota a muchos servicios diferentes de uno o más de nuestros sistemas internos, todos en rápida sucesión. Esto es indicativo de que alguien está haciendo un "rastreo de puertos" de nuestro sistema.

Muchos sistemas modernos usan una combinación de motores de detección del mal uso y del anómalo.

Un segundo nivel de categorización de los sistemas de detección de intrusiones es según su naturaleza:

- Pasiva
- Reactiva

Los sistemas pasivos simplemente detectan la potencial violación de seguridad, registran la información y generan un alerta.

Los sistemas reactivos, por el otro lado, están diseñados para responder ante una actividad ilegal, por ejemplo, sacando al usuario del sistema o mediante la reprogramación de la barrera corta fuego para impedir tráfico

de red desde una fuente presumiblemente hostil. Si bien se podría pensar que un sistema reactivo es la solución ideal, ¿Por qué emplear personal o contratar a alguien cuando la máquina puede efectuar la acción requerida?

8.5 Qué pueden hacer y qué no pueden hacer los Sistemas de Detección de Intrusiones

8.5.1 Pueden:

1. Aumentar el nivel de seguridad general de nuestro entorno.
2. Vigilar el tráfico de red dentro de nuestras barreras corta fuego.
3. Examinar los contenidos de los mensajes de red; por lo tanto, detectando los tipos de ataque, por ejemplo, de "desborde de buffer".
4. Detectar los cambios en archivos y directorios.
5. Detectar tiempos de acceso anormales.

8.5.2 No pueden:

1. Proporcionar una "bala de plata" mágica que elimine todos nuestros problemas de seguridad
2. Reemplazar al personal calificado o la ayuda externa especializada.

8.6 Requisitos de un IDS

Sin importar qué sistemas vigile o su forma de trabajar, cualquier sistema de detección de intrusos ha de cumplir algunas propiedades para poder desarrollar su trabajo correctamente. En primer lugar, y quizás como

característica más importante, el IDS ha de ejecutarse continuamente sin nadie que esté obligado a supervisarlo; independientemente de que al detectar un problema se informe a un operador o se lance una respuesta automática, el funcionamiento habitual no debe implicar interacción con un humano. Podemos fijarnos en que esto parece algo evidente: muy pocas empresas estarían dispuestas a contratar a una o varias personas simplemente para analizar logs o controlar los patrones del tráfico de una red. Sin entrar a juzgar la superioridad de los humanos frente a las máquinas, hemos de tener presente que los sistemas de detección son mecanismos automatizados que se instalan y configuran de forma que su trabajo habitual sea transparente a los operadores del entorno informático.

Otra propiedad, y también como una característica a tener siempre en cuenta, es la aceptabilidad o grado de aceptación del IDS; al igual que sucedía con cualquier modelo de autenticación, los mecanismos de detección de intrusos han de ser aceptables para las personas que trabajan habitualmente en el entorno. Por ejemplo, no ha de introducir una sobrecarga considerable en el sistema, ni generar una cantidad elevada de falsos positivos (detección de intrusiones que realmente no lo son) o de logs, ya que entonces llegará un momento en que nadie se preocupe de comprobar las alertas emitidas por el detector. Por supuesto, si para evitar problemas con las intrusiones simplemente apagamos el equipo o lo desconectamos de la red, tenemos un sistema bastante seguro, pero inaceptable.

Una tercera característica a evaluar a la hora de hablar de sistemas de detección de intrusos es la adaptabilidad del mismo a cambios en el entorno de trabajo. Como todos sabemos, ningún sistema informático puede considerarse estático: desde la aplicación más pequeña hasta el propio kernel, pasando por supuesto por la forma de trabajar de los usuarios, todo cambia con una periodicidad más o menos elevada.

Si nuestros mecanismos de detección de intrusos no son capaces de adaptarse rápidamente a esos cambios, están condenados al fracaso.

Todo IDS debe además presentar cierta tolerancia a fallos o capacidad de respuesta ante situaciones inesperadas; insistiendo en lo que comentábamos antes sobre el carácter altamente dinámico de un entorno informático, algunos de los cambios que se pueden producir en dicho entorno no son graduales sino bruscos, y un IDS ha de ser capaz de responder siempre adecuadamente ante los mismos.

Podemos contemplar, por ejemplo, un reinicio inesperado de varias máquinas o un intento de engaño hacia el IDS; esto último es especialmente crítico: sólo hemos de pararnos a pensar que si un atacante consigue modificar el comportamiento del sistema de detección y el propio sistema no se da cuenta de ello, la intrusión nunca será notificada, con los dos graves problemas que eso implica: aparte de la intrusión en sí, la falsa sensación de seguridad que produce un IDS que no genera ninguna alarma es un grave inconveniente de cara a lograr sistemas seguros.

8.7 Tipos de IDS

8.7.1 HIDS

Como antes hemos comentado, un sistema de detección de intrusos basado en máquina (host-based IDS) es un mecanismo que permite detectar ataques o intrusiones contra la máquina sobre la que se ejecuta.

Tradicionalmente, los modelos de detección basados en máquina han consistido por una parte en la utilización de herramientas automáticas de análisis de logs generados por diferentes aplicaciones o por el propio kernel del sistema operativo, prestando siempre especial atención a los registros relativos a demonios de red, como un servidor Web o el propio inetd, y por otra en el uso de verificadores de integridad de determinados ficheros vitales para el sistema, como el de contraseñas; no obstante, desde hace unos años un tercer esquema de detección se está implantando con cierta fuerza: se trata de los sistemas de detección, honeypots.

El análisis de logs generados por el sistema varía entre diferentes clones de Unix por una sencilla razón: cada uno de ellos guarda la información con un cierto formato, y en determinados ficheros, aunque todos - o casi todos - sean capaces de registrar los mismos datos, que son aquellos que pueden ser indicativos de un ataque. La mayor parte de las versiones de Unix son capaces de registrar casi todas las actividades que

se llevan a cabo en el sistema, en especial aquellas que pueden suponer una vulneración de su seguridad; sin embargo, el problema radica en que pocos administradores se preocupan de revisar con un mínimo de atención esos logs, por lo que muchos ataques contra la máquina, tanto externos como internos, y tanto fallidos como exitosos, pasan finalmente desapercibidos. Aquí es donde entran en juego las herramientas automáticas de análisis, como swatch o logcheck; a grandes rasgos, realizan la misma actividad que podría ejecutar un shellscript convenientemente planificado que incluyera entre sus líneas algunos grep de registros sospechosos en los archivos de log.

¿A qué entradas de estos ficheros debemos estar atentos? Evidentemente, esto depende de cada sistema y de lo que sea "normal" en él, aunque suelen existir registros que en cualquier máquina denotan una actividad cuanto menos sospechosa. Esto incluye ejecuciones fallidas o exitosas, peticiones no habituales al servicio SMTP (como vrfy o expn), conexiones a diferentes puertos rechazadas por TCP Wrappers, intentos de acceso remotos como superusuario, etc; si en la propia máquina tenemos instalado un Firewall independiente del corporativo, o cualquier otro software de seguridad - uno que quizás es especialmente recomendable es PortSentry -, también conviene estar atentos a los logs generados por los mismos, que habitualmente se registran en los ficheros normales de auditoría del sistema (syslog, messages...) y que suelen contener información que con una probabilidad elevada denotan un ataque real.

Por otra parte, la verificación de integridad de archivos se puede realizar a diferentes niveles, cada uno de los cuales ofrece un mayor o menor grado de seguridad. Por ejemplo, un administrador puede programar y planificar un sencillo shellscript para que se ejecute periódicamente y compruebe el propietario y el tamaño de ciertos ficheros como /etc/passwd o /etc/shadow; evidentemente, este esquema es extremadamente débil, ya que si un usuario se limita a cambiar en el archivo correspondiente su UID de 100 a 000, este modelo no descubriría el ataque a pesar de su gravedad. Por tanto, parece obvio que se necesita un esquema de detección mucho más robusto, que compruebe aparte de la integridad de la información registrada en el nodo asociado a cada fichero (fecha de última modificación, propietario, grupo propietario...) la integridad de la información contenida en dicho archivo; y esto se consigue muy fácilmente utilizando funciones resumen sobre cada uno de los ficheros a monitorizar, funciones capaces de generar un hash único para cada contenido de los archivos. De esta forma, cualquier modificación en su contenido generará un resumen diferente, que al ser comparado con el original dará la voz de alarma; esta es la forma de trabajar de Tripwire, el más conocido y utilizado de todos los verificadores de.

Sea cual sea nuestro modelo de verificación, en cualquiera de ellos debemos llevar a cabo inicialmente un paso común: generar una base de datos de referencia contra la que posteriormente compararemos la información de cada archivo. Por ejemplo, si nos limitamos a comprobar

el tamaño de ciertos ficheros debemos, nada más configurar el sistema, registrar todos los nombres y tamaños de los ficheros que deseemos, para después comparar la información que periódicamente registraremos en nuestra máquina con la que hemos almacenado en dicha base de datos; si existen diferencias, podemos encontrarnos ante un indicio de ataque. Lo mismo sucederá si registramos funciones resumen: debemos generar un hash inicial de cada archivo contra el que comparar después la información obtenida en la máquina. Independientemente de los contenidos que deseemos registrar en esa base de datos inicial, siempre hemos de tener presente una cosa: si un pirata consigue modificarla de forma no autorizada, habrá burlado por completo a nuestro sistema de verificación. Así, es vital mantener su integridad; incluso es recomendable utilizar medios de sólo lectura, como un CD-ROM, o incluso unidades extraíbles - discos o disquetes - que habitualmente no estarán disponibles en el sistema, y sólo se utilizarán cuando tengamos que comprobar la integridad de los archivos de la máquina.

Por último, aunque su utilización no esté tan extendida como la de los analizadores de logs o la de los verificadores de integridad, es necesario hablar, dentro de la categoría de los sistemas de detección de intrusos basados en máquina, de los sistemas de decepción o honeypots. Básicamente, estos `tarros de miel' son sistemas completos o parte de los mismos (aplicaciones, servicios, subentornos...) diseñados para recibir ciertos tipos de ataques; cuando sufren uno, los honeypots detectan la actividad hostil y aplican una estrategia de respuesta.

Dicha estrategia puede consistir desde un simple correo electrónico al responsable de la seguridad de la máquina hasta un bloqueo automático de la dirección atacante; incluso muchos de los sistemas - la mayoría - son capaces de simular pirata medianamente experimentado la simulación de vulnerabilidades ha de ser muy "real", dedicando a tal efecto incluso sistemas completos (denominados 'máquinas de sacrificio'), pero con atacantes de nivel medio o bajo dicho engaño es muchísimo más sencillo: en muchos casos basta simular la existencia de un troyano como BackOrifice mediante FakeBO para que el pirata determine que realmente estamos infectados e intente utilizar ese camino en su ataque contra nuestro sistema.

Hemos revisado en este punto las ideas más generales de los sistemas de detección de intrusos basados en host; aunque hoy en día los que vamos a describir a continuación, los basados en red, son con diferencia los más utilizados, como veremos más adelante todos son igualmente necesarios si deseamos crear un esquema de detección con la máxima efectividad. Se trata de niveles de protección diferentes pero que tienen un mismo objetivo: alertar de actividades sospechosas y, en algunos casos, proporcionar una respuesta automática a las mismas.

8.7.2 NIDS

Los sistemas de detección de intrusos basados en red (network-based IDS) son aquellos capaces de detectar ataques contra diferentes sistemas de una misma red (en concreto, de un mismo dominio de

colisión), aunque generalmente se ejecuten en uno solo de los hosts de esa red. Para lograr su objetivo, al menos uno de los interfaces de red de esta máquina sensor trabaja en modo promiscuo, capturando y analizando todas las tramas que pasan por él en busca de patrones indicativos de un ataque.

¿Cuáles pueden ser estos "patrones identificativos de un ataque" a los que estamos haciendo referencia? Casi cualquiera de los diferentes campos de una trama de red TCP/IP puede tener un valor que, con mayor o menor probabilidad, represente un ataque real; los casos más habituales incluyen:

- **Campos de fragmentación** Una cabecera IP contiene dieciséis bits reservados a información sobre el nivel de fragmentación del datagrama; de ellos, uno no se utiliza y trece indican el desplazamiento del fragmento que transportan. Los otros dos bits indican o bien que el paquete no ha de ser fragmentado por un router intermedio o bien que el paquete ha sido fragmentado y no es el último que se va a recibir (MF, More Fragments). Valores incorrectos de parámetros de fragmentación de los datagramas se han venido utilizando típicamente para causar importantes negaciones de servicio a los sistemas y, desde hace también un tiempo incluso para obtener la versión del sistema operativo que se ejecuta en un determinado host; por ejemplo, ¿Qué le sucedería al subsistema de red implantado en el núcleo de una máquina si

nunca recibe una trama con el bit MF reseteado, indicando que es el último de un paquete? se quedaría permanentemente esperándola? ¿Y si recibe uno que en teoría no está fragmentado pero se le indica que no es el último que va a recibir? ¿Cómo respondería el núcleo del sistema operativo en este caso? Como vemos, si en nuestras máquinas observamos ciertas combinaciones de bits relacionados con la fragmentación realmente tenemos motivos para sospechar que alguien trata de atacarnos.

- **Dirección origen y destino** Las direcciones de la máquina que envía un paquete y la de la que lo va a recibir también son campos interesantes de cara a detectar intrusiones en nuestros sistemas o en nuestra red. No tenemos más que pensar el tráfico proveniente de nuestra DMZ que tenga como destino nuestra red protegida: es muy posible que esos paquetes constituyan un intento de violación de nuestra política de seguridad. Otros ejemplos clásicos son las peticiones originadas desde Internet y que tienen como destino máquinas de nuestra organización que no están ofreciendo servicios directos al exterior, como un servidor de bases de datos cuyo acceso está restringido a sistemas de nuestra red.
- **Puerto origen y destino** Los puertos origen y destino son un excelente indicativo de actividades sospechosas en nuestra red. Aparte de los intentos de acceso no autorizado a servicios de nuestros sistemas, pueden detectar actividades que también supondrán a priori violaciones de nuestras políticas de seguridad, como la existencia de troyanos, ciertos tipos de barridos de

puertos, o la presencia de servidores no autorizados dentro de nuestra red.

- **Flags TCP** Uno de los campos de una cabecera TCP contiene seis bits (URG, ACK, PSH, RST, SYN y FIN), cada uno de ellos con una finalidad diferente (por ejemplo, el bit SYN es utilizado para establecer una nueva conexión, mientras que FIN hace justo lo contrario: liberarla). Evidentemente el valor de cada uno de estos bits será 0 o 1, lo cual de forma aislada no suele decir mucho de su emisor; no obstante, ciertas combinaciones de valores suelen ser bastante sospechosas: por ejemplo, una trama con los dos bits de los que hemos hablado - SYN y FIN - activados simultáneamente sería indicativa de una conexión que trata de abrirse y cerrarse al mismo tiempo. Para hacernos una idea de la importancia de estos bits de control, no conviene olvidar que uno de los problemas de seguridad más conocidos de los últimos años sobre plataformas Windows estaba fundamentado básicamente en el manejo de paquetes OOB (Out Of Band): tramas con el bit URG activado.

- **Campo de datos** Seguramente, el campo de datos de un paquete que circula por la red es donde más probabilidades tenemos de localizar un ataque contra nuestros sistemas; esto es debido a que con toda probabilidad nuestro Firewall corporativo detendrá tramas cuya cabecera sea "sospechosa" (por ejemplo, aquellas cuyo origen no esté autorizado a alcanzar su destino o con campos incorrectos), pero rara vez un Firewall se parará a analizar el contenido de los datos transportados en la trama.

Acabamos de ver sólo algunos ejemplos de campos de una trama TCP/IP que, al presentar determinados valores, pueden ser indicativos de un ataque; sin embargo, no todo es tan sencillo como comprobar ciertos parámetros de cada paquete que circula por uno de nuestros segmentos. También es posible y necesario que un detector de intrusos basado en red sea capaz de notificar otros ataques que no se pueden apreciar en una única trama; uno de estos ataques es la presencia de peticiones que, aunque por sí mismas no sean sospechosas, por su repetición en un intervalo de tiempo más o menos pequeño puedan ser indicativas de un ataque (por ejemplo, barridos de puertos horizontales o verticales). Otros ataques difíciles de detectar analizando tramas de forma independiente son las negaciones de servicio distribuidas (DDoS, Distributed Denial of Service), justamente por el gran número de orígenes que el ataque tiene por definición.

Según lo expuesto hasta ahora en este punto, puede parecer que los sistemas de detección de intrusos basados en red funcionan únicamente mediante la detección de patrones; realmente, esto no es así: en principio, un detector de intrusos basado en red puede estar basado en la detección de anomalías, igual que lo puede estar uno basado en máquinas. No obstante, esta aproximación es minoritaria; aunque una intrusión generará probablemente comportamientos anormales (por ejemplo, un tráfico excesivo entre el sistema atacante y el atacado) susceptibles de ser detectados y eliminados, con demasiada frecuencia estos sistemas no detectarán la intrusión hasta que la misma se

encuentre en un estado avanzado. Este problema hace que la mayor parte de IDS basados en red que existen actualmente funcionen siguiendo modelos de detección de usos indebidos.

Para finalizar este punto dedicado a los sistemas de detección de intrusos basados en red es necesario hablar de las honeynets - literalmente, "redes de miel" -. Se trata de un concepto muy parecido al de los honeypots, de los que ya hemos hablado, pero extendido ahora a redes completas: redes diseñadas para ser comprometidas, formadas por sistemas reales de todo tipo que, una vez penetrados, permiten capturar y analizar las acciones que está realizando el atacante para así poder aprender más sobre aspectos como sus técnicas o sus objetivos. Realmente, aunque la idea general sea común, existen dos grandes diferencias de diseño entre un tarro de miel y una honeynet: por un lado, esta última evidentemente es una red completa que alberga diferentes entornos de trabajo, no se trata de una única máquina; por otro, los sistemas dentro de esta red son sistemas reales, en el sentido de que no simulan ninguna vulnerabilidad, sino que ejecutan aplicaciones típicas similares a las que podemos encontrar en cualquier entorno de trabajo "normal". El objetivo de una honeynet no es la decepción, sino principalmente conocer los movimientos de un pirata en entornos semireales, de forma que aspectos como sus vulnerabilidades o sus configuraciones incorrectas se puedan extrapolar a muchos de los sistemas que cualquier empresa posee en la actualidad; de esta forma podemos prevenir nuevos ataques exitosos contra entornos reales.

En el funcionamiento de una "red de miel" existen dos aspectos fundamentales y especialmente críticos, que son los que introducen la gran cantidad trabajo de administración extra que una honeynet implica para cualquier organización. Por un lado, tenemos el control del flujo de los datos: es vital para nuestra seguridad garantizar que una vez que un sistema dentro de la honeynet ha sido penetrado, este no se utilice como plataforma de salto para atacar otras máquinas, ni de nuestra organización ni de cualquier otra; la `red de miel' ha de permanecer perfectamente controlada, y por supuesto aislada del resto de los segmentos de nuestra organización. En segundo lugar, otro aspecto básico es la captura de datos, la monitorización de las actividades que un atacante lleva a cabo en la honeynet. Recordemos que nuestro objetivo principal era conocer los movimientos de la comunidad pirata para poder extrapolarlos a sistemas reales, por lo que también es muy importante para el correcto funcionamiento de una honeynet una correcta recogida de datos generados por el atacante: ha de ser capturada toda la información posible de cada acción, de una forma poco agresiva y por supuesto sin que el atacante se entere. Además, estos datos recogidos nunca se han de mantener dentro del perímetro de la honeynet, ya que si fuera así cualquier pirata podría destruirlos con una probabilidad demasiado elevada.

El concepto de honeynet es relativamente nuevo dentro del mundo de la seguridad y, en concreto, de los sistemas de detección de intrusos; a pesar de ello, se trata de una idea muy interesante que presumiblemente

va a extenderse de una forma más o menos rápida (no todo lo rápida que nos gustaría, ya que implantar y explotar una honeynet no es algo ni trivial, ni mucho menos rápido); cada día más, las herramientas de seguridad no se conforman con detectar problemas conocidos, sino que tratan de anticiparse a nuevas vulnerabilidades que aún no se han publicado pero que pueden estar - y de hecho están - presentes en multitud de sistemas. Conocer cuanto antes cualquier avance de la comunidad underground es algo vital si queremos lograr este objetivo.

Como antes hemos comentado, los sistemas de detección de intrusos basados en red, de los que hemos hablado a lo largo de este punto, son con diferencia los más utilizados actualmente en sistemas en explotación; no obstante, como casi cualquier herramienta relacionada con la seguridad, estos sistemas no son ninguna panacea, y su implantación ha de verse complementada con una correcta configuración de elementos como nuestro Firewall corporativo o, por supuesto, los sistemas de detección basados en host.

Veremos más adelante, en este mismo capítulo, que ambos tipos de IDS son igualmente necesarios en nuestro entorno de trabajo.

IDS	Características
NIDS	▪ Un NIDS escucha constantemente la red en modo promiscuo, en sistemas ▪ Unix casi siempre a través de la interfaz de libpcap. ▪ Herramienta muy poderosa, podemos verificar el tráfico de toda la red con una sóla máquina a la entrada del segmento. ▪ Aplica un primer filtrado para descartar el tráfico que no le interese (hosts/puertos no vigilados). ▪ Analiza el flujo de datos buscando patrones sospechosos basado en firmas ▪ Compara el tráfico entrante contra patrones conocidos. Ágil, extensible, confiable. ▪ Pocos falsos postivos/negativos basado en análisis de anomalías. ▪ Genera un patrón de tráfico no hostil, emplea algún mecanismo para buscar tráfico que no concuerde con este patrón. ▪ Muy poco intrusivo - puede instalarse sin interrumpir la operación de la red, completamente transparente para los sistemas vigilados. ▪ Permite respuesta en tiempo casi-real (vamos: Puede servir para detener un ataque en proceso). ▪ Invisible (o casi invisible en el peor de los casos) para un atacante. ▪ No puede analizar información cifrada. ▪ Típicamente pueden detectar que un ataque fue llevado a cabo, no si fue exitoso o no.
HIDS	▪ Toma su entrada de las bitácoras, registros de auditoría y estado del sistema. ▪ Típicamente, reportan aquellos registros que salen del patrón normal de operación (ya sea autogenerado o definido por el administrador). ▪ Existencia de archivos con SETUID. ▪ Servicios de red inesperados. ▪ Modificaciones en los archivos de configuración del sistema. ▪ Entradas de cron/at. ▪ Binarios cuyo checksum no corresponde al del paquete. ▪ Nuevas cuentas. ▪ Cuentas sin contraseña. ▪ Cuentas que permitan la entrada por intercambio de llaves ssh. ▪ Relaciones de confianza con otros sistemas. ▪ Estado de interfaces de red. ▪ Cambio en el patrón común de consumo de memoria, procesador, red y disco, incluso de horario. ▪ Últimas entradas de los usuarios, especialmente desde hosts poco comunes. ▪ Casi siempre, un HIDS es invocado a intervalos regulares. ▪ Imposible utilizarlo en tiempo real al ocurrir un ataque. ▪ Valiosísimo para entender qué llevó a un ataque Intrusivo. ▪ Puede generar mucho tráfico adicional de red, al enviar reportes a un punto centralizado. ▪ Esto puede delatar su existencia a un atacante. ▪ Nos responde fácilmente si un ataque fue exitoso o no. ▪ Si un atacante sabe de nuestro HIDS, puede que lo primero que haga sea desactivarlo o eliminar sus huellas

Tabla 8.1 Características de los tipos de IDS

8.8 Implementación real de un IDS

Vamos a tratar ahora de definir unas pautas para crear un sistema distribuido de detección de intrusos, capaz de generar respuestas automáticas, alarmas, o simplemente logs a distintos niveles de nuestra arquitectura de red, formando lo que se suele denominar un modelo de seguridad de círculos concéntricos.

Para ello, imaginemos un pirata externo a nuestra organización que intenta atacar una determinada máquina, y pensemos en el primer punto en el que podemos detectar dicho ataque y actuar sobre él: ahí es donde deberemos implantar el primer sensor, ya que se trata de la primera barrera que estamos interponiendo entre el atacante y su objetivo; ese primer punto no es otro que nuestro router de salida a Internet. No podemos entrar aquí a tratar detalles sobre las capacidades de detección de intrusos de productos como los routers Cisco y su IOS, o de otros elementos fácilmente integrables con esta electrónica de red, como NetRanger (también de Cisco), ya que se trata de sistemas que poco tienen que ver con Unix, y que en muchos casos no controlamos nosotros directamente sino una tercera organización (por ejemplo, Telefónica), a pesar de que tengan incluso una dirección IP perteneciente a nuestra red. En las páginas Web de los distintos fabricantes se puede encontrar información muy útil sobre sus productos orientados a la detección y respuesta ante ataques.

8.8.1 IDS en el Firewall

El segundo punto que separará al atacante de su objetivo será el Firewall. Este elemento, sobre el que probablemente ya tendremos pleno control, estará formado por uno o varios sistemas Linux con un software de filtrado de paquetes ejecutándose sobre ellos, y es aquí donde vamos a implantar el primer esquema de detección de intrusos y respuesta automática ante ataques (esta respuesta será habitualmente el bloqueo de la dirección atacante en el propio Firewall).

Para decidir qué tipos de ataques debemos detectar y bloquear en nuestro Firewall debemos pararnos a pensar con qué información trabaja habitualmente este sistema; cualquier Firewall lo hará al menos con los cinco elementos que definen una conexión bajo la pila TCP/IP: dirección origen, dirección destino, puerto origen, puerto destino y protocolo. De estos cinco, quizás los dos menos importantes (de cara a detectar ataques) son quizás el protocolo utilizado y el puerto origen de la conexión; por tanto, son los otros tres elementos los que nos ayudarán en la constitución de nuestro IDS y los que nos facilitarán el poder lanzar una respuesta automática contra el atacante.

Conociendo las direcciones origen y destino y el puerto destino de una conexión ya podemos detectar cierto tipo de ataques; quizás el ejemplo más habitual son los escaneos de puertos, tanto horizontales como verticales, que se lanzan contra nuestros sistemas. La técnica de detección de estos ataques está basada por el momento en comprobar X

eventos de interés dentro de una ventana de tiempo Y. Así, podemos analizar en nuestro Firewall cuándo una misma dirección origen accede a un determinado puerto de varios destinos en menos de un cierto tiempo umbral (escaneo horizontal) o cuando accede a diferentes puertos bien conocidos de un mismo sistema también en menos de ese tiempo umbral (escaneo vertical). ¿Por qué el hecho de fijarnos sólo en puertos bien conocidos en este último caso? Muy sencillo: muchas aplicaciones abren muchos puertos destino, generalmente altos (por encima del 1024), en una única sesión de funcionamiento, por lo que esa sesión sería identificada por el Firewall como un escaneo vertical cuando realmente no lo es.

Una técnica alternativa que con frecuencia suele ser utilizada con bastante efectividad para detectar escaneos verticales consiste en vigilar del acceso a determinados puertos de los sistemas protegidos por el Firewall, acceso que con toda probabilidad representará un intento de violación de nuestras políticas de seguridad. No nos engañemos: si alguien trata de acceder desde fuera del segmento protegido a puertos como echo (7/TCP,UDP), systat (11/TCP), netstat (15/TCP), tcpmux (1/tcp) o el desfasado uucp (540/TCP), lo más probable es que se trate de una persona que no lleva muy buena intención con respecto a nuestras máquinas. Seguramente estará lanzando un escaneo vertical contra nosotros, aunque a veces también se puede tratar de un simple curioso que trata de comprobar nuestro grado de seguridad para lanzar un ataque posterior: evidentemente, si alguien tiene abierto un puerto

de los citados anteriormente, denota una escasa preocupación por su seguridad, por lo que casi con toda certeza se puede intuir que tendrá agujeros importantes en alguna de sus máquinas.

Otro tipo de ataques que también son fácilmente detectables vigilando el acceso a determinados puertos de nuestros sistemas protegidos son aquellos que detectan la presencia - o la comprobación de la presencia - de diferentes troyanos como NetBus o BackOrifice: si en el firewall se detecta tráfico dirigido a puertos como 12345, 12346 o 20034 (TCP) o como 31337 (UDP), sin duda se trata de un atacante que está tratando de aprovechar estos troyanos; en muchos casos - la mayoría - se tratará de escaneos horizontales en busca de máquinas contaminadas a lo largo de toda o gran parte de nuestra clase C.

Servicio	Puerto	Protocolo	Ataque
Ttymux	1	TCP	Escaneo horizontal
Echo	7	TCP/UDP	Escaneo horizontal
Systat	7	TCP	Escaneo horizontal
Daytime	13	TCP/UDP	Escaneo horizontal
Netstat	15	TCP	Escaneo horizontal
Finger	79	TCP	Escaneo horizontal/vertical
Who	513	UDP	Escaneo horizontal
Uucp	540	TCP	Escaneo horizontal/vertical
NetBus	12345	TCP	Troyano
NetBus	12346	TCP	Troyano
NetBus	20034	TCP	Troyano
BackOrifice	31337	UDP	Troyano
Hack´a´Tack	31789	UDP	Troyano
Hack´a´Tack	31790	UDP	Troyano

Tabla 8.2 Algunos puertos a monitorear en un Firewall

En la tabla 8.2 se muestran algunos de los puertos a los que conviene estar atentos a la hora de diseñar una política de detección de intrusos en nuestro Firewall; por supuesto, existen muchos más que pueden ser considerados 'sospechosos', pero en cualquier caso siempre conviene ser muy precavido con su monitorización ya que algunos de ellos pueden ser usados por usuarios lícitos a los que causaríamos una grave negación de servicio si, por ejemplo, les bloqueáramos el acceso a nuestra red a causa de un falso positivo.

Todos sabemos que el Firewall es algo vital para proteger a nuestros sistemas, pero lamentablemente es un elemento muy limitado a la hora de detectar ataques.

Por ejemplo, imaginemos la siguiente situación: un atacante decide comprobar si nuestro servidor Web corporativo tiene algún tipo de vulnerabilidad que le pueda ayudar en un ataque contra la máquina; algo muy común hoy en día, ya que quizás uno de los mayores daños que puede sufrir la imagen de una empresa - especialmente si está relacionada con las nuevas tecnologías - es una modificación de su página Web principal.

Es muy probable que ese pirata lanzara en primer lugar un escaneo de puertos vertical contra el servidor, para comprobar si aparte del servicio HTTP se está ofreciendo algún otro; si todo es correcto, el puerto de Web será el único abierto en el Firewall corporativo, Firewall que además

detectará el ataque contra la máquina y bloqueará, al menos temporalmente, cualquier acceso de la dirección atacante.

Un punto a nuestro favor, pero el pirata no tiene más que colgar el módem y volver a llamar a su proveedor para conseguir otra IP, con lo cual obtiene de nuevo acceso a nuestro servicio HTTP y además ya sabe que el único puerto abierto en la máquina es ese.

Ahora ese atacante no necesita ningún tipo de escaneo de puertos adicional; puede seguir varios caminos para atacarnos, pero sin duda el más lógico y fácil es tratar de localizar vulnerabilidades en el servidor Web de nuestra organización.

Para ello puede lanzar un escaneador de vulnerabilidades en servidores Web contra la máquina, escaneador que no generará ninguna alerta en el Firewall; al fin y al cabo, lo único que hacen estos programas es lanzar peticiones al puerto 80 de nuestro servidor Web, algo que el Firewall no contempla como sospechoso: para él, no hay diferencia entre un analizador de CGIs y peticiones normales a nuestras páginas.

Parece por tanto evidente que nuestra primera barrera de detección de intrusos es realmente útil, pero también insuficiente frente a determinados ataques; entonces entra en juego el segundo nivel de nuestro sistema de detección de intrusos, el ubicado en el segmento de

red - en el dominio de colisión - en el que se encuentra el host que estamos tratando de proteger.

8.8.2 IDS en la red

Debemos tener cuidado cuando decidimos en qué momento implementar nuestro nuevo sistema de detección. Analicemos el siguiente ejemplo de red.

Si colocamos un NIDS fuera de nuestra barrera corta fuego externa, ganaremos una ventaja con el aviso prematuro, dado que permitiría detectar el rastreo de puertos de reconocimiento que, típicamente, señala el comienzo de una actividad de hacking. Sin embargo, no todos los rastreos serán seguidos por un real ataque, dado que el hacker puede determinar que no tenemos en estos momentos debilidades que él pueda explotar.

Esto podría llevarnos a una numerosa cantidad de alertas que no requerirán de nuestra atención. Una consecuencia común y perjudicial de esto es que los involucrados pierden confianza en el IDS y comienzan a ignorar los alertas.

Podríamos tomar la opción de usar nuestra barrera corta fuego externa para alertarnos del tráfico que ha sido negado y colocar nuestro IDS en nuestro DMZ (De-Militarized Zone). Una ventaja de esto es que podemos adecuar nuestra base de datos de firmas atacantes del NIDS a fin de

considerar únicamente aquellos ataques que corresponden a los sistemas que están en DMZ; nuestra barrera corta fuego bloqueará todo el tráfico restante.

Podríamos tener interés en el tráfico no autorizado dirigido a nuestra red privada. Un NIDS ubicado dentro de la red privada podría monitorear todo el tráfico para, desde y dentro de esa red. Un sistema de ese tipo no tendría que ser tan poderoso como, por ejemplo, un NIDS localizado en la parte exterior de nuestra barrera corta fuego externa, dado que tanto el volumen como el tipo de tráfico que es necesario controlar se reduce enormemente.

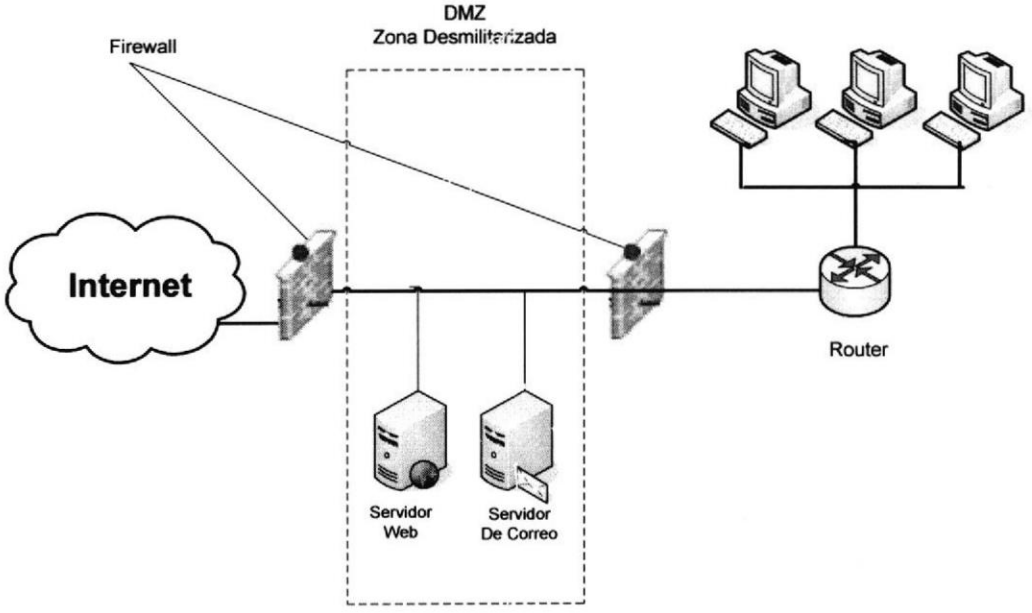


Figura 8.1 Estableciendo la zona desmilitarizada

8.8.2.1 Redes conmutadas

NIDS trabaja buceando en todo el flujo de tráfico que pasa por la red. Si simplemente conectamos el NIDS a un puerto normal de una red conmutada, será muy poco útil. La propia naturaleza de las redes conmutadas implica que solamente el tráfico destinado a un determinado dispositivo, es el que se envía al mismo. Existen dos maneras de enfocar esta situación.

Una alternativa es conectar el NIDS a un puerto "medidor" sobre un conmutador apropiado. Un puerto medidor es aquel que está programado para recibir copias de todo el tráfico que fluye a través del conmutador o un sub-juego seleccionado del mismo. Con este enfoque existe el problema que el ancho de banda acumulado que demanda el NIDS podría llegar a ser demasiado impactante.

Una segunda opción es usar un dispositivo de interrupción de la red. Esto permite captar los paquetes que viajan en un sentido desde la conexión, pero es solamente aconsejable para la intercepción del tráfico dirigido a un solo dispositivo de la red.

8.8.2.2 Juntando todas las cosas

El diagrama que aparece más abajo muestra una configuración de red común. El IDS1 de Red, vigila el tráfico proveniente desde Internet. Por lo tanto, buscará los mensajes de correo o de la Web tramposos.

El IDS2 de Red vigila todo el tráfico que pasa hacia y desde la red interna.

El tercer sistema NIDS vigila específicamente todo el tráfico desde y hacia la red de Recursos Humanos. DMZ y los otros sistemas individuales estarían funcionando sobre un software de detección de intrusiones basado en host, configurado en función del perfil de riesgos de cada caso.

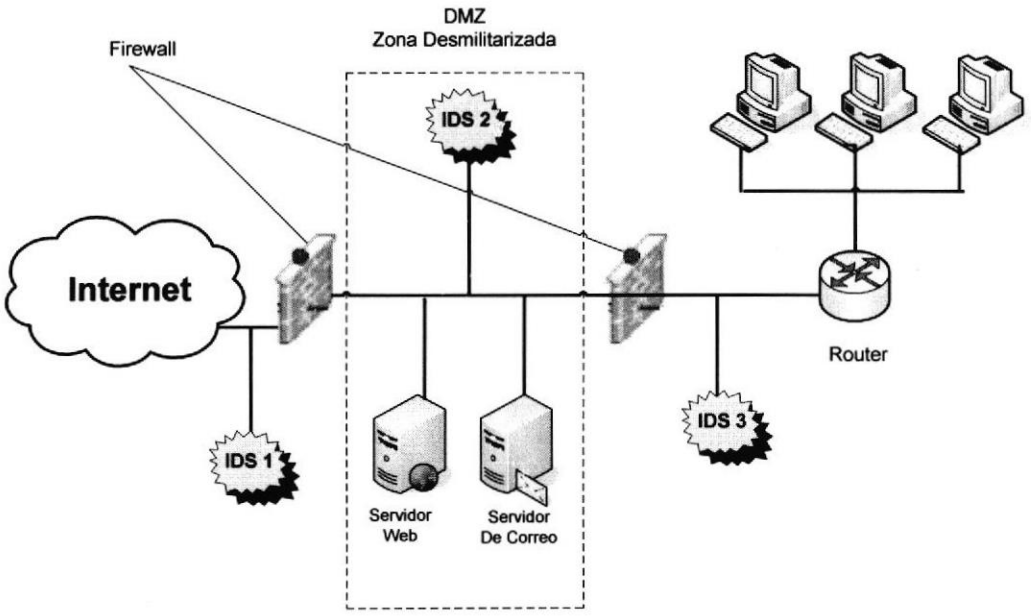


Figura 8.2 Ejemplo 1

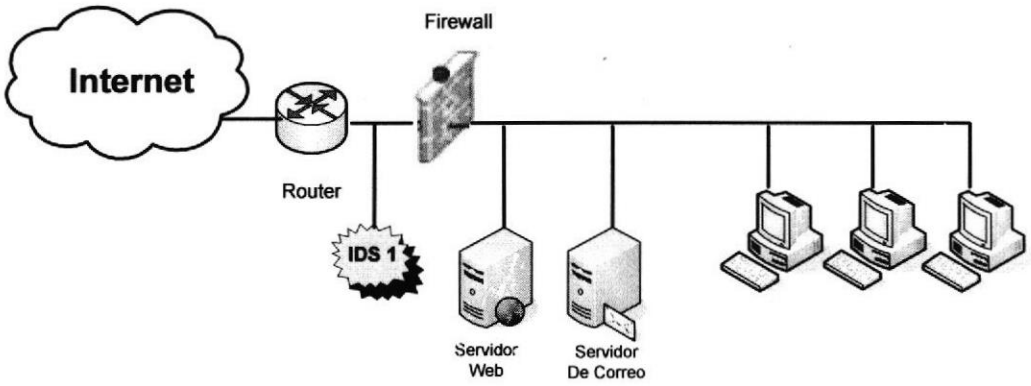


Figura 3.3 Ejemplo 2

8.8.3 Cisco IDS

El sistema de la detección de intrusos de Cisco fue diseñado para proteger eficientemente su infraestructura de los datos y de la información. Con la complejidad creciente de las amenazas de la seguridad, la seguridad eficiente de intrusos en la red es crítica. La protección vigilante asegura continuidad del negocio y reduce al mínimo los efectos de intrusiones costosas.

Los adelantos de los IDS de Cisco incluyen cuatro elementos esenciales que cuando están combinados proporcionen una solución segura, eficiente, y comprensiva de la protección de la intrusión:

8.8.3.1 Detección exacta de la amenaza

Como el elemento central en la lista de sistema de la detección de intrusos de Cisco, la versión del programa del sensor de las

identificaciones del Cisco 4.x proporciona seguridad sin precedente contra las amenazas sabidas y desconocidas que apuntan su red, incluyendo gusanos, ataques del negación de servicios (DOS), y ataques del uso. Las ayudas de las IDS 4.x del Cisco aseguran cobertura comprensiva empleando métodos de detección múltiples y proporcionando la capacidad para prevenir la ejecución de ataques detectados. También ofrece varias características de empleo fácil integradas para maximizar eficacia.

8.8.3.2 Protección Comprensiva De la Amenaza

- **Métodos de detección múltiples** - métodos múltiples de las aplicaciones de las IDS 4.x del Cisco para detectar exactamente amenazas, incluyendo el reconocimiento de patrón, el análisis del protocolo, la detección de la anomalía del tráfico, y la detección de la anomalía del protocolo. Además, las IDS del Cisco entregan un motor de la firma de la capa 2 para proporcionar la protección contra técnicas spoofing del Address Resolution Protocol (ARP).
- **Protocolo extenso que supervisa** - todos los protocolos importantes de TCP/IP se supervisan, incluyendo el IP, el Internet Control Message Protocol (ICMP), el TCP, y el User Datagram Protocol (UDP). Las IDS 4.x del Cisco también descifran protocolos de capa de uso, tales como ftp, Simple Mail Transfer Protocol (smtp), HTTP, Domain Name System (DNS), llamada del alejado-procedimiento (RPC), NetBIOS, protocolo del transporte de las noticias de la red (NNTP), telnet, y (P2P).

- **Detección comprensiva del ataque** - las IDS 4.x del Cisco tienen las capacidades más comprensivas de la detección en las categorías siguientes:

  - Actividad de la explotación indica tentativas de acceso o de sistemas de la red promiscuos.
  - Actividad del DOS indica tentativas de consumir anchura de banda o de computar recursos para interrumpir operaciones normales.
  - Actividad del reconocimiento indica tentativas de sondear su red para identificar blancos, tales como barridos del puerto.
  - Actividad de mal empleo indica tentativas de violar la política corporativa; detectado configurando el sensor para buscar secuencias de texto de encargo en el tráfico de la red.

8.8.3.3 Investigación inteligente de la amenaza

La tecnología de la respuesta de la amenaza del Cisco elimina virtualmente alarmar falsos y se determina automáticamente qué amenazas necesitan la atención inmediata evitar una intrusión costosa.

Como parte del IDS de Cisco el Treta Response es una tecnología que ayuda al proveer una eficiente solución de protección contra intrusos. Esta inteligente tecnología virtualmente elimina las falsas alarmas, y de esta forma reduce costos de intrusiones.

A diferencia de otras soluciones que manejan intrusiones, solo la tecnología Treta Response de Cisco provee un automatizado análisis en tiempo real de cada host objetivo para determinar que ha ocurrido y como ha sido direccionada. ¿El resultado? Falsas alarmas son eliminadas e intrusiones reales son inmediatamente identificadas, salvando el tiempo, recursos y los altos costos asociados.

8.8.3.4 Fácil de administrar

Las herramientas Browser-basadas simplifican la interacción del usuario mientras que proporcionan las herramientas analíticas de gran alcance que ayudan a respuestas rápidas y eficientes a las amenazas.

Cisco proporciona la seguridad eficaz que supervisa y configura sin importar el tamaño del despliegue, usando una gama de opciones de administración.

Todas las herramientas de administración se diseñan con un GUI intuitivo y la navegación fácil, permite la instalación, la configuración, y el monitoreo rápido de los acontecimientos y de los dispositivos de la seguridad.

8.8.4 Dragon de Enterasys

Dragon está compuesto por tres productos, Dragon Sensor, Dragon Squire y Dragon Server. Cada uno de los tres productos que

configuran Dragon 5 presenta unas funciones diferentes. Por un lado, Dragon Sensor es un sistema de detección de intrusiones basado en la red de alta velocidad, mientras que Dragon Squire se emplea para detener los ataques que se producen en los hosts locales y Dragon Server agrupa todas las características del producto mediante la gestión y generación de informes centralizados.

8.8.4.1 Descripción de la defensa de la Intrusión

A pesar de esfuerzos casi universales de proteger redes corporativas, las organizaciones distribuidas de hoy siguen siendo susceptibles a una multiplicidad de ataques.

Es necesario ampliar seguridad más allá de la espina dorsal corporativa para proteger una variedad de vulnerabilidades potenciales, incluyendo las conexiones del Internet, canales de comunicaciones entre las oficinas alejadas y corporativas y los acoplamientos entre los socios de negocio confiados.

Desafortunadamente, muchas medidas preventivas empleadas para asegurar recursos corporativos y el tráfico interno proporcionan la anchura o la profundidad del análisis necesitado para identificar y para responder a los ataques procurados o para destapar amenazas potenciales a través de la organización. Los Firewalls que despliegan o las redes privadas virtuales, por ejemplo, pueden reducir al mínimo la exposición, pero no proporcionan simplemente bastante protección.

8.8.4.2 ¿Por qué Dragón?

El dragón de Enterasys resuelve el desafío de asegurar una red moderna proporcionando los sensores de velocidad altos para detectar y para responder a la actividad sospechosa, a los datos forenses para determinar el impacto de los ataques de la red, y la escalabilidad para desplegar y para manejar una gran cantidad de sensores sin negativamente la afectación de la operación de su red.

De la energía combinada de su sensor de la red, del sensor del anfitrión, el dragón proporciona una solución integrada para detectar y parar uso erróneo y ataques a través de la infraestructura.

Combina acontecimientos en la red con ésos en los anfitriones, los Firewall, las rebajadoras, los interruptores, proporcionando la detección completa para el ambiente pequeño y grande. Con análisis comprensivo y capacidades de supervisión en tiempo real, el dragón entrega un nivel más alto de la defensa a la infraestructura del negocio.

8.8.4.3 Características de valor añadido

- Gerencia escalable integrada para la detección de la intrusión, y la solución integrada disponible.
- Software flexible y solución para los anfitriones de Windows, de Linux, de Solaris, de AIX y de HPUX.

- Administración avanzada de la política y centralizada que supervisa, analiza y divulga.
- Controla hackers en sus ataques.
- Arquitectura altamente escalable para adaptarse a cualquier empresa.
- Supervisión de la seguridad.
- Estadísticas extensas de la información.
- Alta visibilidad en el estado de la red con el tiempo real y un histórico de procesos.
- Reportes a nivel ejecutivo, para la interpretación fácil, sin importar la maestría técnica del usuario.
- Detección de método múltiple incluyendo la concordancia con el modelo, descifrado del protocolo, y la detección de la anomalía.

8.8.5 Etrust

El detecto de intrusos eTrust es una solución completa que incorpora tres claves de seguridad un paquete, un manejo comprensivo de la red, un sistema de prevención, monitoreo y filtros de Internet en tiempo real. Estas soluciones trabajan juntas para dirigir requerimientos específicos de seguridad, formando una red a la defensa sin altos costos.

8.8.5.1 Soluciones seguras de la confianza de Etrust

Las soluciones de la gerencia de la seguridad de la confianza de CA's e alinean las ediciones de seguridad de su organización con sus

necesidades del negocio automatizando, simplificando y aerodinamizando procesos y proporcionando visibilidad en tiempo real en la multiplicidad de los acontecimientos de la seguridad que ocurren diariamente - permitir la respuesta derecha en el tiempo derecho.

Las soluciones de la gerencia de la seguridad de la confianza de CA's e se agrupan en tres áreas: e confía en identidad y tiene acceso a la gerencia, a la gerencia de la amenaza de la confianza de e, y a la gerencia de información de la seguridad de la confianza de e.

8.8.5.2 Gerencia de la amenaza de la confianza de Etrust

Las soluciones de la gerencia de la amenaza de la confianza de CA's e proporcionan un acercamiento integrado a la seguridad, permitiéndole alinear a la gerencia de la amenaza con su negocio. Las soluciones de la gerencia de la amenaza de la confianza que entregan la protección del virus, control del Spam, URL que se filtra, supervisión del secreto de los datos, gerencia de la vulnerabilidad, detección de la intrusión, verificación de la integridad del sistema operativo de z/OS y revisando, y más. Además, nos permiten ayudar a identificar y a amenazar la seguridad antes de que afecten su negocio.

8.8.6 Snort

Si volvemos a la clasificación de IDS que hemos presentado al principio de este capítulo, podemos clasificar a SNORT como un sistema basado

en red (se monitoriza todo un dominio de colisión) y que funciona mediante detección de usos indebidos. Estos usos indebidos se reflejan en una base de datos formada por patrones de ataques; dicha base de datos se puede descargar también desde la propia página Web de SNORT[1], donde además se pueden generar bases de patrones `a medida' de diferentes entornos (por ejemplo, ataques contra servidores Web, intentos de negaciones de servicio, exploits...). El archivo que utilicemos en nuestro entorno será la base para el correcto funcionamiento de nuestro sistema de detección de intrusos.

Para instalar un sistema de detección de intrusos basado en SNORT en primer lugar necesitamos evidentemente este programa, que podemos descargar desde su página Web. Además, para compilarlo correctamente es necesario disponer de las librerías libpcap, un interfaz para tratamiento de paquetes de red desde espacio de usuario, y es recomendable también (aunque no obligatorio) instalar Libnet, librería para la construcción y el manejo de paquetes de red. Con este software correctamente instalado en nuestro sistema, la compilación de SNORT es trivial.

Una vez hemos compilado e instalado correctamente el programa llega el momento de ponerlo en funcionamiento; y es aquí donde se produce uno de los errores más graves en la detección de intrusos. Por lógica,

[1] www.snort.org

uno tiende a pensar que el sensor proporcionará mejores resultados cuantos más patrones de ataques contenga en su base de datos; nada más lejos de la realidad.

En primer lugar, es muy probable que no todos los ataques que SNORT es capaz de detectar sean susceptibles de producirse en el segmento de red monitorizado; si situamos el sensor en una zona desmilitarizada donde únicamente ofrecemos servicio de Web, ¿qué interés tiene tratar de detectar ataques contra DNS?

Lo lógico es que las políticas implementadas en nuestro Firewall ni siquiera dejen pasar tráfico hacia puertos que no sean los de los servidores Web pero, incluso en caso de que el potencial ataque se produjera entre máquinas del propio segmento, hemos de evaluar con mucho cuidado si realmente vale la pena sobrecargar la base de datos con patrones que permitan detectar estos ataques. Evidentemente, cuanta más azúcar más dulce, pero si el sensor ha de analizar todo el tráfico, quizás mientras trata de decidir si un paquete entre dos máquinas protegidas se adapta a un patrón estamos dejando pasar tramas provenientes del exterior que realmente representan ataques: hemos de tener presente que el sniffer no detendrá el tráfico que no sea capaz de analizar para hacerlo más tarde, sino que simplemente lo dejará pasar. Así, debemos introducir en la base de patrones de ataques los justos para detectar actividades sospechosas contra nuestra red.

En segundo lugar, pero no menos importante, es necesario estudiar los patrones de tráfico que circulan por el segmento donde el sensor escucha para detectar falsos positivos y, o bien reconfigurar la base de datos, o bien eliminar los patrones que generan esas falsas alarmas. Aunque suene algo crudo, si un patrón nos genera un número considerable de falsos positivos, debemos plantearnos su eliminación: simplemente no podremos decidir si se trata de verdaderas o de falsas alarmas. Esto es especialmente crítico si lanzamos respuestas automáticas contra las direcciones `atacantes' (por ejemplo, detener todo su tráfico en nuestro Firewall): volviendo al ejemplo de la zona desmilitarizada con servidores Web, podemos llegar al extremo de detener a simples visitantes de nuestras páginas simplemente porque han generado falsos positivos; aunque en un entorno de alta seguridad quizás vale la pena detener muchas acciones no dañinas con tal de bloquear también algunos ataques (aunque constituiría una negación de servicio en toda regla contra los usuarios que hacen uso legítimo de nuestros sistemas), en un entorno normal de producción esto es impensable. Seguramente será más provechoso detectar y detener estos ataques por otros mecanismos ajenos al sensor.

En resumen, hemos de adaptar a nuestro entorno de trabajo, de una forma muy fina, la base de datos de patrones de posibles ataques. Vale invertir el tiempo que sea necesario en esta parte de la implantación, ya que eso nos ahorrará después muchos análisis de falsas alarmas y, por qué negarlo, algún que otro susto; una vez tengamos todo configurado,

podemos utilizar el siguiente script para lanzar SNORT de forma automática al arrancar el sistema:

```
anita:~# cat /etc/init.c/snort
#!/sbin/sh
#
# Instalacion:
#       # cp <script> /etc/init.d/snort
#       # chmod 744 /etc/init.d/snort
#       # chown root:sys /etc/init.d/snort
#       # ln /etc/init.d/snort /etc/rc2.d/S99snort
#

# Directorio de log
DIRLOG=/var/log/snort
# Fichero de reglas
RULES=/usr/local/security/snort.conf
# Ejecutable
SNORT=/usr/local/security/snort
# Interfaz
IF=hme0

case "$1" in
'start')
        if [ ! -d "$DIRLOG" ]; then
                mkdir -p "$DIRLOG"
        fi
        if [ ! -r "$RULES" ]; then
                echo "No puedo leer el fichero de
patrones..."
                exit -1
        fi
        if [ ! -x "$SNORT" ]; then
                echo "No encuentro el ejecutable..."
                exit -1
        fi
        $SNORT -l $DIRLOG -c $RULES -i $IF -N -D
        ;;
'stop')
        if [ ! -r "/var/run/snort_$IF.pid" ]; then
                echo "No puedo obtener el PID..."
                exit -1
        fi
        kill -TERM `cat /var/run/snort_$IF.pid`
        ;;
*)
        echo "Usage: $0 { start | stop }"
        exit 1
esac
exit 0
anita:~#
```

Con el sensor y sus patrones correctamente configurados ya estamos listos para poner en funcionamiento nuestro sistema de detección de intrusos. Seguramente hasta ahora no hemos tenido muchos problemas con el IDS; no obstante, a partir de ahora las cosas se empiezan a complicar un poco, ya que comienza la segunda parte, la del tratamiento de la información que nuestro sensor nos va a proporcionar. Y es que desde este momento el sistema de detección va a empezar a funcionar y a generar logs con notificaciones de posibles ataques, o cuanto menos de actividades sospechosas; es hora de decidir cosas como dónde situar al sensor, qué hacer ante la generación de un evento en el mismo, cómo procesar la información recibida, o simplemente cuándo rotar los logs generados.

El último de los problemas planteados realmente tiene fácil solución; ¿Cuándo rotar los logs que SNORT genera? La respuesta es muy sencilla: depende. Depende de la cantidad de informes generados en nuestro sensor, depende de la frecuencia con la que debamos realizar informes de los ataque sufridos, depende de la implementación elegida para ejecutar respuestas automáticas ante un ataque, etc. En definitiva, la rotación correcta de unos logs es algo que se debe estudiar y planificar para cada entorno concreto, no se puede dar un periodo estricto que se aplique siempre porque sería sin duda erróneo. No obstante, una idea que nos puede ayudar en la toma de esta decisión es la siguiente: rotaremos los logs cuando los hayamos procesado y extraído de ellos la información que nos pueda interesar para proteger nuestro entorno.

SNORT genera logs en el directorio /var/log/snort/ si no le indicamos lo contrario. En ese directorio encontraremos un fichero denominado alert con las actividades que se vayan registrando, y, si no hubiéramos especificado la opción '-N' al arrancar el programa, una serie de subdirectorios cuyos nombres son las direcciones IP de las máquinas de las que se detecta alguna actividad (es el denominado 'packet logging'). Como nosotros lo que buscamos es básicamente la generación de alarmas, independiente del packet logging, no necesitamos generar estos directorios (aunque nada nos impide hacerlo).

El siguiente shellscript planificado convenientemente con crontab (si lo ejecutamos más de una vez durante el día quizás nos interese afinar la variable $FECHA) puede ser utilizado para realizar la rotación del archivo de alarmas generado por SNORT:

```
anita:~# cat /usr/local/security/rotalog
#!/bin/sh
#
# Directorio de log
DIRLOG=/var/log/snort
# Fecha (DD/MM/YY)
FECHA=`date +%d.%m.%Y`
# Interfaz
IF=hme0

if [ ! -d "$DIRLOG" ]; then
        mkdir -p "$DIRLOG"
fi
cd $DIRLOG
mv alert alert-$FECHA
touch alert
chmod 600 alert
kill -HUP `cat /var/run/snort_$IF.pid`
compress alert-$FECHA
anita:~#
```

Independientemente de la rotación de logs que llevemos a cabo en cada sensor, suele resultar interesante centralizar todos los logs generados en un sólo sistema (a veces se le denomina maestro o master), aunque sólo sea para realizar estadísticas, seguimientos de máquinas atacantes y atacadas, o simplemente un `top ten' de piratas. Para ello podemos establecer relaciones de confianza entre los sensores y ese maestro para que puedan conectarse entre sí sin necesidad de contraseñas y, de forma automática, transferir los logs almacenados y rotados. Por supuesto, a estas alturas dicha relación no la estableceremos mediante la definición de máquinas confiables en archivos .rhosts o similares, ni con las herramientas r-, sino mediante SSH y las claves públicas y privadas de cada máquina. Aparte de una mayor seguridad (no autenticamos a una máquina simplemente por su dirección o nombre, algo fácilmente falseable), siguiendo un mecanismo de este estilo conseguimos que todas las comunicaciones entre sistemas se realicen de forma cifrada, algo que aquí es muy importante: cualquier información relativa a potenciales ataques o respuestas automáticas a los mismos se ha de considerar como confidencial, por lo que sería un grave error echar todo nuestro trabajo a perder simplemente porque alguien sea capaz de esnifar dicho tráfico.

Volviendo a nuestras cuestiones iniciales, también debíamos decidir dónde situar lógicamente al sensor; por ejemplo, una cuestión típica es si debemos emplazarlo detrás o delante del Firewall que protege a

nuestra red. En principio, si dejamos que el sensor analice el tráfico antes de que sea filtrado en el Firewall, estaremos en disposición de detectar todos los ataques reales que se lanzan contra nuestra red, sin ningún tipo de filtrado que pueda detener las actividades de un pirata; no obstante, probablemente lo que más nos interesará no es detectar todos estos intentos de ataque (aunque nunca está de más permanecer informado en este sentido), sino detectar el tráfico sospechoso que atraviesa nuestro Firewall y que puede comprometer a nuestros servidores. Por tanto, es recomendable reemplazar el sensor de nuestro sistema de detección de intrusos en la zona protegida; de cualquier forma, los potenciales ataques que no lleguen al mismo quedarán registrados en los logs del Firewall, e incluso serán neutralizados en el mismo.

Como el sensor ha de analizar todo el tráfico dirigido a las máquinas protegidas, si nos encontramos en un entorno donde dichas máquinas se conecten mediante un concentrador (hub) o mediante otras arquitecturas en las que cualquiera de ellas vea (o pueda ver) el tráfico de las demás, no hay muchos problemas de decisión sobre dónde situar al sensor: lo haremos en cualquier parte del segmento. Sin embargo, si nuestros sistemas se conectan con un switch la cuestión se complica un poco, ya que en las bocas de este elemento se verá únicamente el tráfico dirigido a las máquinas que estén conectadas a cada una de ellas; en este caso, tenemos varias opciones. Una de ellas puede ser modificar por completo nuestra arquitectura de red para

integrar un concentrador por el que pasen los paquetes ya filtrados antes de llegar a las máquinas del switch. No obstante, suelen existir alternativas más sencillas y cómodas, como la replicación de puertos que se puede configurar en la mayoría de switches; la idea es muy simple: todo el tráfico dirigido a determinada boca del switch se monitoriza y se duplica en otra boca.

Así, no tenemos más que configurar este port mirroring y replicar la boca por la que se dirige el tráfico hacia el segmento de máquinas a monitorizar, enviándolo también a una segunda boca en la que conectaremos nuestro sensor.

Para acabar con los comentarios sobre dónde y cómo situar al sensor de nuestro sistema detector de intrusos, un último apunte: nos conviene recordar que la interfaz por la que se analiza el tráfico no tiene por qué tener dirección IP. Perfectamente podemos tener un interfaz levantado e inicializado pero sin asignarle ninguna dirección.

Esto nos puede resultar útil si no nos interesa que en el segmento protegido se detecte una nueva máquina, o simplemente si no queremos que nuestro sensor sea alcanzable de alguna forma por el resto de sistemas de su dominio de colisión. Para nuestra comodidad (por ejemplo, a la hora de centralizar logs de diferentes sensores) podemos usar una máquina con dos interfaces, una escuchando todo el

tráfico y la otra configurada de forma normal, que será por la que accedamos al sistema.

8.8.6.1 Características de SNORT

En esta versión encontramos:

- Multi-acontecimiento agregado que hace cola en Snort. Snort ahora apoya acontecimientos múltiples de registro por paquete, y da la prioridad a esos acontecimientos usando diversos métodos.
- Detecta problemas fijos del timezone con los plugins de la salida de la base de datos.
- Incluye solamente la funcionalidad en línea de la base. Esto significa que los regla-tipos de la GOTA, de SDROP, y del RECHAZO están apoyados.
- La detección plugin "substituye" también se ha incluido en esta versión.
- Actualizaciones y verificación del estado en línea y de los paquetes que caen malas sumas de comprobación.
- Incluye un motor portscan nuevo de la detección - sfPortscan. Este motor fue desarrollado para detectar protocolo de TCP/UDP/ICMP/I.
- Además de esto, detecta la trampa y portscans distribuidos, y puede distinguir entre las exploraciones filtradas y sin filtro. Cuando se generan las alarmas portscan, los detalles del portscan se registran junto con él. Esta información da al analista los

detalles en cuántos puertos fueron explorados, las gamas, número de IPS explorado, gamas del IP, y qué puertos estaban abiertos.

- Incluye varios arreglos del ChangeLog para la información adicional.

8.9 Conclusión

Los IDS (Sistemas de Detección de Intrusiones) aportan una capacidad precautoria anticipada a sus defensas, alertándolo acerca de toda actividad sospechosa que, típicamente, ocurre antes y durante un ataque.

Dado que, la mayoría de las veces, no se puede detener un ataque, los sistemas de detección de intrusiones no deberían ser tenidos en cuenta como una alternativa a las buenas medidas tradicionales de seguridad. No existen sustitutos para una cuidadosamente pensada política de seguridad empresarial, respaldada por procedimientos eficaces que sean ejecutados por personal calificado utilizando las herramientas adecuadas. Los sistemas de detección de intrusiones deberían visualizarse como una herramienta adicional en la batalla continua contra hackers y crackers.

Actualmente, ya están disponibles una gran variedad de sistemas de detección de intrusiones, adecuados para casi todas las circunstancias. La oferta pasa desde versiones gratuitas que pueden instalarse en PC de bajo porte hasta sistemas comerciales que cuestan miles de dólares y que demandan lo último y lo más poderoso en hardware. Algunos están diseñados para monitorear redes completas, mientras otros están

destinados a implementarse en máquina por máquina. Como era de esperar, todos tienen pros y contras y existe un rol para cada uno de ellos. Pero entre ellos tenemos el Snort que es un buen IDS fácil de encontrar ya que es gratuito presta la seguridad necesaria a bajo costo.

Una solución de seguridad no es algo que se pueda definir como un producto que se instala o se aplica y está terminado, la seguridad es un proceso constante de estudio de nuevos métodos de ataque, servicios, parches y productos que están disponibles en Internet para atacar redes, sistemas, y aplicaciones. Es un proceso de auditorias constantes, de actualización de políticas y procedimientos preventivos ante posibles ataques.

CAPÍTULO 9

9. ESTABLECIENDO POLÍTICAS DE ACCESO Y DE SEGURIDAD

9.1 Introducción

En la actualidad, las organizaciones son cada vez más dependientes de sus redes informáticas y un problema que las afecte, por mínimo que sea, puede llegar a comprometer la continuidad de las operaciones.

La falta de medidas de seguridad en las redes es un problema que está en crecimiento. Cada vez es mayor el número de atacantes y cada vez están más organizados, por lo que van adquiriendo día a día habilidades más especializadas que les permiten obtener mayores beneficios. Tampoco deben subestimarse las fallas de seguridad provenientes del interior mismo de la organización.

La propia complejidad de la red es una dificultad para la detección y corrección de los múltiples y variados problemas de seguridad que van apareciendo. En medio de esta variedad, han ido aumentando las acciones poco respetuosas de la privacidad y de la propiedad de recursos y sistemas. "Hackers", "crakers", entre otros, han hecho aparición en el vocabulario ordinario de los usuarios y de los administradores de las redes

Además de las técnicas y herramientas criptográficas, es importante recalcar que un componente muy importante para la protección de los sistemas consiste en la atención y vigilancia continua y sistemática por parte de los responsables de la red.

A la hora de plantearse en qué elementos del sistema se deben de ubicar los servicios de seguridad podrían distinguirse dos tendencias principales:

9.2 La seguridad

En la actualidad, la seguridad informática ha adquirido gran auge, dadas las cambiantes condiciones y las nuevas plataformas de computación disponibles.

La posibilidad de interconectarse a través de redes, ha abierto nuevos horizontes que permiten explorar más allá de las fronteras de la organización. Esta situación ha llevado a la aparición de nuevas amenazas en los sistemas computarizados.

Consecuentemente, muchas organizaciones gubernamentales y no gubernamentales internacionales han desarrollado documentos y directrices que orientan en el uso adecuado de estas destrezas tecnológicas y recomendaciones con el objeto de obtener el mayor provecho de estas ventajas, y evitar el uso indebido de la mismas. Esto puede ocasionar serios problemas en los bienes y servicios de las empresas en el mundo.

En este sentido, las políticas de seguridad informática (PSI) surgen como una herramienta organizacional para concientizar a cada uno de los miembros de una organización sobre la importancia y la sensibilidad de la información y servicios críticos que favorecen el desarrollo de la organización y su buen funcionamiento.

9.2.1 ¿Cuál puede ser el valor de los datos?

Establecer el valor de los datos es algo totalmente relativo, pues la información constituye un recurso que, en muchos casos, no se valora adecuadamente debido a su intangibilidad, cosa que no ocurre con los equipos, la documentación o las aplicaciones.

Además, las medidas de seguridad no influyen en la productividad del sistema por lo que las organizaciones son reticentes a dedicar recursos a esta tarea.

Cuando hablamos del valor de la información nos referimos, por ejemplo, a qué tan peligroso es enviar la información de mi tarjeta de crédito a través de Internet para hacer una compra, en una red gigantesca donde viajan no únicamente los 16 dígitos de mi tarjeta de crédito sino millones de datos más , gráficas, voz y vídeo.

En efecto, el tema no está restringido únicamente a Internet. Aunque no se esté conectado a Internet, una red está expuesta a distintos tipos de ataques electrónicos, incluidos los virus. Por esto, y por cualquier otro

tipo de consideración que se tenga en mente, es realmente válido pensar que cualquier organización que trabaje con computadoras especialmente con servidores Web debe tener normativas que hacen al buen uso de los recursos y de los contenidos, es decir, al buen uso de la información.

9.2.2 Implementación de medidas de seguridad

La implementación de medidas de seguridad, es un proceso técnico- administrativo. Como este proceso debe abarcar toda la organización, sin exclusión alguna, ha de estar fuertemente apoyado por el sector gerencial, ya que sin ese apoyo, las medidas que se tomen no tendrán la fuerza necesaria.

Hay que tener muy en cuenta la complejidad que suma a la operatoria de la organización la implementación de estas medidas.

Será necesario sopesar cuidadosamente la ganancia en seguridad respecto de los costos administrativos y técnicos que se generen.

Después de analizar todo lo expuesto anteriormente, resulta claro que proponer o identificar una política de seguridad requiere de un alto compromiso con la organización, agudeza técnica para establecer fallas y debilidades, y constancia para renovar y actualizar la política establecida en función del dinámico ambiente que rodea las organizaciones modernas.

9.3 Políticas generales de seguridad

9.3.1 Elementos de una política de seguridad informática

Como mencionábamos en el apartado anterior, una PSI debe orientar las decisiones que se toman en relación con la seguridad. Por tanto, requiere de una disposición por parte de cada uno de los miembros de la empresa para lograr una visión conjunta de lo que se considera importante.

Objetivos de la política y descripción clara de los elementos involucrados en su definición.

- Responsabilidades por cada uno de los servicios y recursos informáticos a todos los niveles de la organización.
- Requerimientos mínimos para configuración de la seguridad de los sistemas que cobija el alcance de la política.
- Definición de violaciones y de las consecuencias del no cumplimiento de la política.
- Responsabilidades de los usuarios con respecto a la información a la que ella tiene acceso.

Las PSI deben ofrecer explicaciones comprensibles acerca de por qué deben tomarse ciertas decisiones, transmitir por qué son importantes estos u otros recursos o servicios.

De igual forma, las PSI establecen las expectativas de la organización en relación con la seguridad y lo que ella puede esperar de las acciones que la materializan en la compañía.

Deben mantener un lenguaje común libre de tecnicismos y términos legales que impidan una comprensión clara de las mismas, sin sacrificar su precisión y formalidad dentro de la empresa.

Por otra parte, la política debe especificar la autoridad que debe hacer que las cosas ocurran, el rango de los correctivos y sus actuaciones que permitan dar indicaciones sobre la clase de sanciones que se puedan imponer. No debe especificar con exactitud qué pasara o cuándo algo sucederá; no es una sentencia obligatoria de la ley.

Finalmente, las Políticas de Seguridad Informática como documentos dinámicos de la organización, deben seguir un proceso de actualización periódica sujeto a los cambios organizacionales relevantes: crecimiento de la planta de personal, cambio en la infraestructura computacional, alta rotación de personal, desarrollo de nuevos servicios, cambio o diversificación de negocios entre otros.

9.3.2 Algunos parámetros para establecer PSI

Si bien las características de la PSI que hemos mencionado hasta el momento, nos muestran una perspectiva de las implicaciones en la

formulación de estas directrices, revisaremos a continuación, algunos aspectos generales recomendados para la formulación de las mismas.

- Considere efectuar un ejercicio de análisis de riesgos informático, a través del cual valore sus activos, el cual le permitirá afinar las PSI de su Web Server.
- Involucre a las áreas propietarias de los recursos o servicios, pues ellos poseen la experiencia y son fuente principal para establecer el alcance y las definiciones de violaciones a la Política de Seguridad de Información.
- Comunique a todo el personal involucrado en el desarrollo de las PSI, los beneficios y riesgos relacionados con los recursos y bienes, y sus elementos de seguridad.
- Desarrolle un proceso de monitoreo periódico de las directrices.

Un último consejo:

No dé por hecho algo que es obvio. Haga explícito y concreto los alcances y propuestas de seguridad analizadas, con el propósito de evitar sorpresas y malos entendidos en el momento de establecer los mecanismos de seguridad que respondan a las políticas de seguridad trazadas.

9.3.3 Proposición de una forma de realizar el análisis para llevar a cabo un sistema de seguridad informática

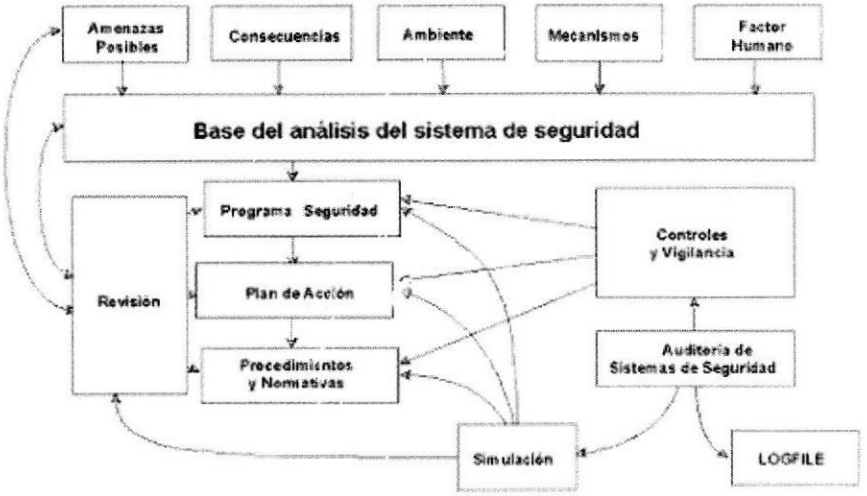


Figura 9.1 Diagrama para el análisis de un sistema de seguridad

Tal como puede visualizarse, en el gráfico están plasmados todos los elementos que intervienen para el estudio de una política de seguridad.

Se comienza realizando una evaluación del factor humano interviniente, teniendo en cuenta que éste es el punto más vulnerable en toda la cadena de seguridad, de los mecanismos con que se cuentan para llevar a cabo los procesos necesarios, luego, el medio ambiente en que se desempeña el sistema, las consecuencias que puede traer aparejado defectos en la seguridad, y cuáles son las amenazas posibles.

Una vez evaluado todo lo anterior, se origina un programa de seguridad, que involucra los pasos a tomar para poder asegurar el umbral de seguridad que se desea. Luego, se pasa al plan de acción, que es cómo se va a llevar a cabo el programa de seguridad. Finalmente, se redactan los procedimientos y normas que permiten llegar a buen destino.

Con el propósito de asegurar el cumplimiento de todo lo anterior, se realizan los controles y la vigilancia que aseguran el fiel cumplimiento de los tres puntos antepuestos.

Para asegurar un marco efectivo, se realizan auditorías a los controles y a los archivos logísticos que se generen en los procesos implementados (de nada vale tener archivos logísticos si nunca se los analizan o se los analizan cuando ya ha ocurrido un problema).

Con el objeto de confirmar el buen funcionamiento de lo creado, se procede a simular eventos que atenten contra la seguridad del sistema. Como el proceso de seguridad es un proceso dinámico, es necesario realizar revisiones al programa de seguridad, al plan de acción y a los procedimientos y normas.

Estas revisiones, tendrán efecto sobre los puntos tratados en el primer párrafo y, de esta manera, el proceso se vuelve a repetir. Es claro que el establecimiento de políticas de seguridad es un proceso dinámico sobre el que hay que estar actuando permanentemente, de manera tal

que no quede sin actualizar; que, cuando se le descubran debilidades, éstas sean subsanadas.

9.3.4 ¿Por qué las políticas de seguridad informática generalmente no consiguen implantarse?

Muchas veces, las organizaciones realizan grandes esfuerzos para definir sus directrices de seguridad y concretarlas en documentos que orienten las acciones de las mismas, con relativo éxito. Según algunos estudios resulta una labor ardua convencer a los altos ejecutivos de la necesidad de buenas políticas y prácticas de seguridad informática.

Muchos de los inconvenientes se inician por los tecnicismos informáticos y por la falta de una estrategia de mercadeo de los especialistas en seguridad que, llevan a los altos directivos a pensamientos como: "más dinero para los juguetes de los ingenieros".

Esta situación ha llevado a que muchas empresas con activos muy importantes, se encuentren expuestas a graves problemas de seguridad que, en muchos de los casos, lleva a comprometer su información sensible y por ende su imagen corporativa.

Ante esta encrucijada, los encargados de la seguridad deben asegurarse de que las personas relevantes entienden los asuntos importantes de la seguridad, conocen sus alcances y están de acuerdo con las decisiones tomadas en relación con esos asuntos. En particular, la gente debe

conocer las consecuencias de sus decisiones, incluyendo lo mejor y lo peor que podría ocurrir. Una intrusión o una travesura puede convertir a las personas que no entendieron, en blanco de las políticas o en señuelos de los verdaderos vándalos. Y estos y comportamientos nos debe llevar a reconocer las pautas de seguridad necesarias y suficientes que aseguren confiabilidad en las operaciones y funcionalidad.

Algoritmo

Cuando se piensa establecer una estrategia de seguridad, la pregunta que se realiza, en primera instancia, es: ¿en qué baso mi estrategia?. La respuesta a esta pregunta es bien simple. El algoritmo Productor/Consumidor.

En este algoritmo, hay dos grandes entidades: una que es la encargada de producir la información; la otra entidad es el consumidor de esta información y otra, llamada precisamente "otros".

Entre el productor y el consumidor, se define una relación que tiene como objetivo una transferencia de "algo" entre ambos, sin otra cosa que intervenga en el proceso. Si esto se logra llevar a cabo y se mantiene a lo largo del tiempo, se estará en presencia de un sistema seguro.

En la realidad, existen entidades y/o eventos que provocan alteraciones a este modelo.

El estudio de la seguridad, en pocas palabras, se basa en la determinación, análisis y soluciones de las alteraciones a este modelo. En una observación y planteo del modelo, determinamos que sólo existen cuatro tipos de alteraciones en la relación producción-consumidor (ver el gráfico del algoritmo).

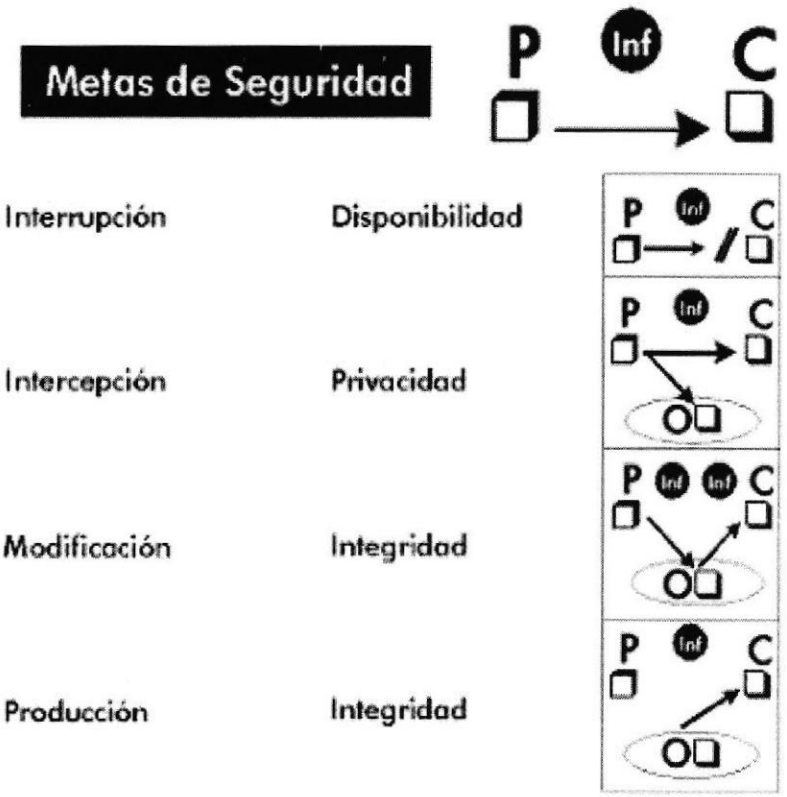


Figura 9.2 Algoritmo del Productor – Consumidor de la información[1]

[1] Manual de seguridad/ Claudia E. Bello

9.3.5 Las PSI como base de la Administración de seguridad

Las políticas de seguridad informática conforman el conjunto de lineamientos que una organización debe seguir para asegurar la confiabilidad de sus sistemas.

Las políticas de seguridad informática son parte del engranaje del sistema de seguridad que la organización posee para salvaguardar sus activos.

Las Políticas de Seguridad Informática constituyen las alarmas y compromisos compartidos en la organización, que le permiten actuar proactivamente ante situaciones que comprometan su integridad. Por tanto, deben constituir un proceso continuo y retroalimentado que observe la concientización, métodos de acceso a la información, monitoreo de cumplimiento y renovación, aceptación de las directrices y estrategia de implantación, que lleven a una formulación de directivas institucionales que logren aceptación general.

Las políticas por sí solas no constituyen una garantía para la seguridad de la organización, ellas deben responder a intereses y necesidades organizacionales basadas en la visión de negocio, que lleven a un esfuerzo conjunto de sus actores por administrar sus recursos, y a reconocer en los mecanismos de seguridad informática factores que facilitan la formalización y materialización de los compromisos adquiridos con la organización.

9.3.6 Plan de Seguridad

Una vez identificados los problemas generales llega la pregunta que supone el principal escollo para desarrollar un plan que corrija la situación: ¿cómo se debe abordar la seguridad en la organización?

El Plan de Seguridad debe ser un proyecto que desarrolle los objetivos de seguridad a largo plazo de la organización, siguiendo el ciclo de vida completo desde la definición hasta la implementación y revisión.
La forma adecuada para plantear la planificación de la seguridad en una organización debe partir siempre de la definición de una política de seguridad que defina el QUÉ se quiere hacer en materia de seguridad en la organización para a partir de ella decidir mediante un adecuado plan de implementación el CÓMO se alcanzarán en la practica los objetivos fijados.

La Política de Seguridad englobará pues los objetivos, conductas, normas y métodos de actuación y distribución de responsabilidades y actuará como documento de requisitos para la implementación de los mecanismos de seguridad.

La política debe contemplar al menos la definición de funciones de seguridad, la realización de un análisis de riesgos, la definición de normativa y procedimientos, la definición de planes de contingencia ante desastres y la definición del plan de auditoría.

A partir de la Política de Seguridad se podrá definir el Plan de Implementación, que es muy dependiente de las decisiones tomadas en ella, en el que se contemplará: el estudio de soluciones, la selección de herramientas, la asignación de recursos y el estudio de viabilidad.

Hay dos cuestiones fundamentales que deberán tenerse en cuenta para implantar con éxito una política de seguridad: Es necesario que la política sea aprobada para que este respaldada por la autoridad necesaria que asegure su cumplimiento y la asignación de recursos; y es necesario que se realicen revisiones periódicas que la mantengan siempre actualizada y acorde con la situación real del entorno.

La Política de Seguridad y el Plan de Implementación (y la implantación propiamente dicha) están íntimamente relacionados:

La Política de Seguridad define el Plan de Implementación ya que la implementación debe ser un fiel reflejo de los procedimientos y normas establecidos en la política.

El Plan de Seguridad debe estar revisado para adaptarse a las nuevas necesidades del entorno, los servicios que vayan apareciendo y a las aportaciones que usuarios, administradores, etc. vayan proponiendo en función de su experiencia. La revisión es esencial para evitar la obsolescencia de la política debido al propio crecimiento y evolución de la organización. Los plazos de revisión deben estar fijados y permitir

además revisiones extraordinarias en función de determinados eventos (por ejemplo, incidentes). El Plan de Implementación debe ser auditado para asegurar la adecuación con las normas.

El Plan de Implementación debe realimentar a la Política de Seguridad. La experiencia, los problemas de implantación, las limitaciones y los avances tecnológicos, etc. permitirán que la política pueda adecuarse a la realidad, evitando la inoperancia por ser demasiado utópica y la mejora cuando el progreso lo permita. Un enfoque como el propuesto asegurará la adecuación del nivel de seguridad implantado con las necesidades de la organización y el correcto seguimiento y control de los riesgos.

Sistema Métrico de Seguridad	PLAN SMS Básico	PLAN SMS Auditor	PLAN SMS ProActivo	PLAN LOPD	PLAN LSSI	PLAN SMS Legal
1º Estudio de acceso a la Información. 2º Estudio de configuración Servicios y Aplicaciones. 3º Estudio de infraestructuras de red y transmisiones. 4º Tests de vulnerabilidad y análisis de Estudios.	X	X	X			X
5º Elaboración de Plan de Seguridad		X	X			X
6º Establecimiento de parámetros de Medición.			X			X
7º Evaluación de parámetros y reajustes de seguridad.			X			X
1º Informe de identificación de "Datos de carácter personal". 2º Inscripción de ficheros en el registro general de Protección de datos. 3º Elaboración de Normas obligatorias de seguridad. 4º Implantación de medidas correctoras.				X		X
1º Informe de identificación de Servicios regulados por la ley. 2º Elaboración de informe de seguridad para servicios regulados. 3º Implantación de sistemas y procedimientos de adecuación.					X	X

Figura 9.3 Cuadro de Planes de Seguridad

9.4 Políticas

9.4.1 Políticas de Seguridad que no pueden pasar por alto

1. La política del impedir. Se puede tomar la política de prohibir. Es decir, podemos poner barreras para impedir el paso a ciertas partes conocidas como sensibles. Pensemos en barreras de control como nombre de usuario/contraseña en los sistema LINUX; muros de seguridad (firewalls) en las redes, Tener todo bajo llave y sólo después de verificar minuciosamente la identidad del usuario y sus

intenciones, dejarle utilizar a éste los recursos sensibles. Esta política tiene la ventaja de la detección temprana de problemas. Esto es, la alternativa es similar a tener todo bajo llave en el almacén y tener que mostrar su credencial o identificación para entrar. Esto sería la política de la prohibición. El inconveniente fundamental es la afectación sensible de la sencillez, rapidez y servicios que ofrece este sistema. La operación se torna burocrática, y por consiguiente pierde facilidad de respuesta a circunstancias diversas. Es conocido que cuando algo extraordinario ocurre, es necesario no tener en cuenta los controles de acceso. El cuerpo de salvamento y de bomberos nunca respetan los controles de acceso!!

2. La política de vigilar. También podemos tomar la política de vigilar todas las actividades o en su defecto, sólo las actividades sospechosas o más sensibles. Permitir el acceso "libre" a todo usuario de los recursos del sistema tiene la ventaja de la libre circulación e intercambio de información, rapidez, sencillez y un gran número de servicios. El inconveniente fundamental es el costo de los supervisores (programas en constante ejecución) y el peligro más latente de encontrarse con problemas y fallas. A este respecto, podemos indicar que la programación en shell, awk y perl son herramientas indispensables para la construcción de los demonios o supervisores en UNIX. Tener todo a la vista y al alcance, pero con agentes supervisores por todos lados del almacén, o sólo sobre todo en los artículos más costosos y con mayor riesgo de extraerse sin pagar.

3. Política prohibir/supervisar. Dejar entrar al almacén libremente, pero pedir credencial de identificación a ciertos departamentos de éste. Al interior de este departamento con información sensible, se tienen agentes supervisores. Esta es una mezcla de la prohibición y supervisión, donde en ciertas zonas ponemos controles de acceso, pero estando dentro, sólo se supervisan algunas cuantas actividades. Cada una de estas políticas tiene sus ventajas e inconvenientes. La ventaja de esta política híbrida prohibir/supervisar es que permite la libre circulación en casi todos los departamentos y sólo analiza con detalle los movimientos en la zona sensible, limitando de esta forma los controles. El inconveniente es un mayor riesgo en las zonas de libre circulación.
4. Cifrado total o selectivo de información. En esta política dejamos la libre circulación por los recu sos del sistema, ya que aun en el caso de realizarse un copia no permitida de datos, la información será inutilizable ya que se encuentra cifrada. La ventaja mayor es la libre circulación de la información que permite un mayor flujo y distribución. Sin embargo, su desventaja es el desconocimiento y poca utilización en los corporativos de las técnicas de cifrado. Los mecanismos de cifrado todavía no son estudiados regularmente en las escuelas y es necesario que se establezca una cultura, primero dentro de los centros educativos y en los corporativos. Si hoy en día solicitamos personal con conocimientos de cifrado de información, no se encontrará. Sin embargo, podemos solicitar personal con conocimientos de matemáticas y seguramente después de una etapa

corta de adaptación se tendrá un experto en estas técnicas, que como se podrán imaginar son 99% asociadas a matemáticas.

La información es un bien a cuidar que se encuentra almacenada en un sistema operativo. Por lo que se debe tener en cuenta los controles de acceso ofrece el sistema operativo. A este respecto es conveniente mencionar que el sistema operativo UNIX es muy recomendable.

La información se encuentra almacenada en una máquina, que hoy en día en general se encuentra conectada en red con otras. Hoy en el mundo de redes mundiales de información al estilo Internet hay dos aspectos en la balanza. Por un lado tenemos la compartición de información en forma rápida, sencilla a un gran conglomerado de usuarios. Por el otro tenemos la seguridad y resguardo de la misma.

Según dicen todos los expertos, la máquina más segura es la que está apagada.

Una política de seguridad: poner guardianes a la entrada

Así como la política de seguridad en las colonias residenciales es no tener ubicada la mansión del dueño a la orilla de la vía, en el mundo de la informática y en las redes de computadoras, la idea es no tener la máquina con la información vital directamente conectada a la red. Se desea generar una subred con una máquina como puente o compuerta y la máquina con información vital en la subred. A este mecanismo en

ocasiones se le llama máquina firewall o máquina muro contra incendios. Un sistema muro de seguridad ("firewall") reemplaza a un ruteador IP con un nodo con varias tarjetas que no permite el avance de paquetes.

En un ruteador todos los paquetes se dirigen hacia adelante a través de la capa IP. Con un sistema firewall no se permite el avance de paquetes. Cada paquete dirigido al sistema firewall se procesa localmente por éste. La diferencia entre ruteador y máquina muro de seguridad radica en que la primera solo verifica la dirección de destino del paquete e indica adecuadamente el camino que tomará éste.

Por el contrario, la máquina analiza todo el contenido del paquete y si se desea realiza una copia de éste en una bitácora y después realiza la acción del ruteador. Es decir, es un ruteador que analiza el contenido del paquete. Una máquina muro de seguridad es un instrumento dedicado a esta actividad. Por este motivo, recientemente se están ofreciendo máquinas con esta característica.

9.4.2 Seguridad en redes

Hoy en día en las empresas se tienen muchas máquinas conectadas en red con el objetivo de poder compartir recursos físicos y lógicos entre todos los usuarios. Una red es una vía donde circulan datos cuyo objetivo es ofrecer a todos los nodos que la forman, un acceso a todos los recursos físicos y lógicos.

La seguridad en la red es un medio de control del acceso y resguardo a estos recursos.

Así, se quiere dar acceso fácil a todo, pero teniendo control sobre todo. Por lo que, entre el grado de acceso y la seguridad se debe establecer un compromiso. Es decir, estos dos temas están en una balanza.

El sistema más seguro, es el que no comparte nada. Es sistema más vulnerable, es aquél que todo ofrece sin saber a quién.

El problema de las computadoras, las redes que las unen y la seguridad de los datos, es análogo al problema de los pueblos, las vías de comunicación y las barreras o controles que se establecen para protección de los pueblos.

Si no tenemos carreteras y nadie externo puede llegar al pueblo, es seguro que nadie externo nos atacará; sólo nos queda cuidar a los agentes internos. Esto no permite el intercambio, que es una gran fuente de ideas y recursos. Por lo que, para establecer una política de seguridad se recomienda construir carreteras y permitir un libre tráfico, pero con verificación de los elementos que desean entrar en el pueblo.
Es decir, no se cierran las carreteras para verificar quién circula; se ponen controles al momento de acceso al pueblo, y después dentro del pueblo se ponen controles estrictos en los centros y actividades neurálgicos o vitales para el desarrollo del pueblo.

Es decir, no se trata de asociar un policía a cada visitante, sino de establecer una bitácora en cada punto vital y que constantemente se esté analizando para detectar los puntos que están siendo atacados o con grandes posibilidades de que lo sean. Por ejemplo, en los bancos las bitácoras son las cámaras de filmación, los detectores de movimiento y paso de objetos por medio de rayos infrarrojos en zonas vitales, etcétera. Es decir, no impido el acceso al banco a los clientes que quieren hacer operaciones bancarias, sino los superviso. Otro ejemplo, en los procesos electorales no se impide el acceso a la casilla, sino se supervisa el comportamiento de los elementos neurálgicos que pueden cambiar el resultado.

Aquí un nodo sería una casilla, el conjunto de casillas la red y el Centro de acopio de todas las urnas, sería el equivalente a la Gran Máquina que administra, controla y orquesta a todos los nodos. Aquí no se cuidan la calles que permiten la circulación entre los nodos.

Se requiere instrumentar políticas de control y supervisión en los nodos.

En conclusión, la seguridad en la red se relaciona sobre todo con la seguridad en cada nodo y no directamente en la red.

9.4.3 Políticas de Seguridad del Servidor

Para que un servidor sea seguro, tanto el Sistema Operativo como la aplicación servidor deben ser seguros, reforzado además por las políticas que el Administrador de ese servidor configure.

Debemos de partir del siguiente hecho: No hay servidor 100% seguro.
Son demasiados los factores que pueden afectar la seguridad, tanto físicos como operativos y electrónicos:

- Desastres naturales
- Fallas de corriente eléctrica
- Fallas de Hardware por deterioro (uso y mal uso)
- Accidentes por mala manipulación de los equipos
- Mala configuración de los equipos
- Mala instalación de los equipos
- Deficientes políticas de seguridad
- Deficiente configuración del software
- Ignorancia
- Software con problemas de seguridad

Un administrador de red debe estar bien enterado de todas las deficiencias de seguridad posibles en los sistemas que administra.
Es su responsabilidad crear las políticas necesarias para que no existan riesgos por parte de los usuarios, configurar bien los servidores para que

no existan riesgos por parte de intrusos, y mantener actualizados sus sistemas contra cualquier ataque conocido.

En los últimos años han proliferado los ataques de tipo Denial Of Service, donde un hacker satura un servidor hasta el punto que deja de "servir". Y de la misma manera, existen programas que olfatean el trafico para detectar un posible ataque de este tipo.

Aquellos servidores conectados a redes publicas (Internet, conexiones telefónicas, WAN, etc.) son los mas vulnerables, puesto que están expuestos a cualquier usuario a cualquier hora y desde cualquier lugar. Es aquí donde la Auditoría de Sistemas toma tanta importancia, como una herramienta forense. Es responsabilidad del Administrador indicar cuales son las deficiencias, cuales pueden ser corregidas, y que medidas hay que tomar para aquellas que no pueden ser corregidas; incluso, hablar de un cambio de plataforma para dar una solución adecuada al problema.

Siempre Alerta

Sin importar la plataforma, para mantener la seguridad de un servidor web hay que estar siempre alerta a los ataques y posibles riesgos, establecer políticas y procedimientos para evitarlos y superarlos, supervisar los registros de auditoría, mantenerse al día con las actualizaciones para los servicios en uso, y siempre estar preparados para lo peor.

Ninguna medida está de más. La clave es tener en mente que, si un hacker realmente se propone entrar al servidor, tarde o temprano lo hará. Una buena configuración y políticas tan sólo le hará el trabajo mucho más difícil. Un buen respaldo constante asegurará que el efecto de la intrusión no afecte tanto al cliente.

9.4.4 Propósitos de las políticas de seguridad

El propósito de esta política es establecer normas para la configuración baja de equipo del servidor interior que se posee y/o se opera por la empresa. La aplicación eficaz de esta política minimizará acceso desautorizado a la empresa, la información propietario y tecnología.

9.4.5 Propiedad y Responsabilidad

Todos los servidores interiores desplegaron a <el Nombre de la Compañía> debe ser poseído por un grupo operacional que es responsable para la administración del sistema. Deben establecerse guías de configuración de servidor aceptado y deben ser mantenidas por cada grupo operacional, debe ser basado en necesidades de negocio y debe ser aprobado por InfoSec.

Los grupos operacionales deben supervisar complacencia de la configuración y deben llevar a cabo una política de la excepción entallada a su ambiente. Cada grupo operacional debe establecer un proceso por

cambiar la configuración guía que incluye revisión y aprobación por InfoSec.

Deben registrarse Servidores de dentro del sistema de dirección de empresa corporativo. A un mínimo, la información siguiente se exige identificar el punto de contacto positivamente:

Servidor y situación, y un contacto del backup, Hardware y Sistema / Versión Operando, las funciones Principales y aplicaciones, si aplicable. Debe guardarse Información dentro del sistema de dirección de empresa corporativo moderno. La Configuración cambia para los servidores de la producción debe seguir los procedimientos de dirección de cambio apropiados.

9.4.6 Pautas de la configuración

La configuración del sistema operativo debe estar de acuerdo con las pautas de InfoSec aceptado.

- Deben desactivarse los Servicios y aplicaciones que no se usarán.
- Acceso a los servicios deben protegerse a través de los métodos del acceso-mando como Envolturas de TCP, si es posible.
- Los más recientes parches de seguridad deben instalarse en el sistema, la única excepción es cuando la aplicación inmediata interferiría con requisitos comerciales.

- Confianza en las relaciones entre los sistemas son un riesgo de seguridad, y su uso debe evitarse. No use una relación verdadera cuando algún otro método de comunicación lo haga.
- Establecer principios de seguridad de acceso requerido a realizar una función.
- Si una metodología para la conexión segura está disponible (es decir, técnicamente factible), debe realizarse acceso privilegiado encima de las conexiones seguras, (ej., los métodos encriptados conectan una red de computadoras conexiones que usan SSH o IPSec).
- Deben localizarse a los Servidores en un ambiente que se controle el acceso físicamente.

9.4.7 Auditando las políticas de seguridad

- Las Auditorías se realizarán en una base regular por organizaciones autorizadas dentro de la compañía.
- Las Auditorías serán manejadas por el grupo de la auditoría interior, de acuerdo con la Política de la Auditoría.
- Cada esfuerzo permitirá impedir a las auditorías causar fracasos operacionales o rupturas.

CAPÍTULO 10

10. DESARROLLAR PROCEDIMIENTOS PARA UN CORRECTO MANTENIMIENTO

10.1 Acerca de los procedimientos

Si se piensa certificar ISO, es indispensable tener un manual de procedimientos escrito y llevarlo a cabo al pie de la letra. De esta manera, cabría pensar que un manual de procedimientos es un paso adelante para poder llegar a la certificación ISO.

10.1.1 Procedimiento de alta de cuenta de usuario

Cuando un elemento de la organización requiere una cuenta de operación en el sistema, debe llenar un formulario que contenga, al menos los siguientes datos:

- Nombre y Apellido
- Puesto de trabajo
- Jefe inmediato superior que avale el pedido
- Descripción de los trabajos que debe realizar en el sistema

Consentimiento de que sus actividades son susceptibles de ser auditadas en cualquier momento y de que conoce las normas de "buen uso de los recursos" (para lo cual, se le debe dar una copia de tales normas).
Explicaciones breves, pero claras de cómo elegir su password.

Asimismo, este formulario debe tener otros elementos que conciernen a la parte de ejecución del área encargada de dar de alta la cuenta, datos como:

- Tipo de cuenta
- Fecha de caducidad
- Fecha de expiración
- Datos referentes a los permisos de acceso (por ejemplo, tipos de permisos a los diferentes directorios y/o archivos)

Si tiene o no restricciones horarias para el uso de algunos recursos y/o para el ingreso al sistema.

10.1.2 Procedimiento de baja de cuenta de usuario

Este procedimiento es el que se lleva a cabo cuando se aleja un elemento de la organización o cuando alguien deja de trabajar por un determinado tiempo. En base a la explicación anterior hay, entonces, dos tipos de alejamientos: permanente y parcial. Aquí, es necesario definir un circuito administrativo a seguir, y que como todos los

componentes de la política de seguridad, debe estar fuertemente apoyado por la parte gerencial de la organización.

Un ejemplo de este circuito, podría ser: ante el alejamiento de un elemento de la organización, la gerencia de personal (o la sección encargada de la administración de los RRHH), debe informar en un formulario de "Alejamiento de personal", todos los datos del individuo que ha dejado la organización, así como de la posición que éste ocupaba y el tipo de alejamiento (permanente o no). Una vez llegada la información al departamento encargado de la administración de sistemas, se utiliza para dar de baja o inhabilitar la cuenta del usuario.

La definición de si se da de baja o se inhabilita es algo importante pues, si se da de baja, se deberían guardar y eliminar los archivos y directorios del usuario, mientras que si sólo se inhabilita, no pasa de esa acción. Si el alejamiento del individuo no era permanente, al volver a la organización, la sección que había informado anteriormente de la ausencia, debe comunicar su regreso, por medio de un formulario dando cuenta de tal hecho para volver a habilitar la cuenta al individuo.

10.1.3 Procedimiento para determinar password

Aunque no lo parezca, la verificación de palabras claves efectivas no es algo frecuente en casi ninguna organización. El procedimiento debe explicar las normas para elegir una password. No tiene que tener relación directa con las características del usuario. Debe constar de

caracteres alfanuméricos, mayúsculas, minúsculas, números y símbolos de puntuación. Determinar, si es posible, el seguimiento de las palabras claves (llevar registros de las palabras claves anteriores elegidas por el usuario). Una vez que el usuario ha elegido su password, se le debe correr un "programa crackeador" para tener idea de cuán segura es, en base al tiempo que tarda en romper la palabra.

10.1.4 Procedimientos de verificación de accesos

Debe explicar la forma de realizar las auditorías de los archivos logísticos de ingresos a fin de detectar actividades anómalas. También debe detectar el tiempo entre la auditoría y cómo actuar en caso de detectar desviaciones. Normalmente, este trabajo es realizado por programas a los que se les dan normativas de qué y cómo comparar. Escanean archivos de "log" con diferentes fechas tomando en cuenta las reglas que se le han dado. Ante la detección de un desvío, generan reportes informando el mismo.

En el procedimiento debe quedar perfectamente indicado quién es el responsable del mantenimiento de estos programas y cómo se actúa cuando se generan alarmas.

10.1.5 Procedimiento para el chequeo del tráfico de la red

Permite conocer el comportamiento del tráfico en la red, al detectar variaciones que pueden ser síntoma de mal uso de la misma. El procedimiento debe indicar el/los programas que se ejecuten, con qué

intervalos, con qué reglas de trabajo, quién se encarga de procesar y/o monitorear los datos generados por ellos y cómo se actúa en consecuencia.

10.1.6 Procedimiento para el monitoreo de los volúmenes de correo

Este procedimiento permite conocer los volúmenes del tráfico de correo o la cantidad de "mails" en tránsito. Dicho procedimiento se encuentra realizado por programas que llevan las estadísticas, generando reportes con la información pedida.

El conocimiento de esta información permite conocer, entre otras cosas, el uso de los medios de comunicación, y si el servidor está siendo objeto de un "spam".

Como en los casos anteriores, en el procedimiento debe estar explicitado quién es el encargado del mantenimiento y del análisis de los datos generados, y qué hacer cuando se detectan variaciones.

10.1.7 Procedimientos para el monitoreo de conexiones activas

Este procedimiento se efectúa con el objeto de prevenir que algún usuario deje su terminal abierta y sea posible que alguien use su cuenta.

El procedimiento es ejecutado por medio de programas que monitorean la actividad de las conexiones de usuarios.

Cuando detecta que una terminal tiene cierto tiempo inactiva, cierra la conexión y genera un log con el acontecimiento.

10.1.8 Procedimiento de modificación de archivos

Este procedimiento sirve para detectar la modificación no autorizada y la integridad de los archivos y, en muchos casos, permite la traza de las modificaciones realizadas. Al igual que en los casos anteriores, debe estar bien determinada la responsabilidad de quién es el encargado del seguimiento y de actuar en caso de alarmas.

10.1.9 Procedimientos para el resguardo de copias de seguridad

Este procedimiento debe indicar claramente dónde se deben guardar las copias de seguridad y los pasos a seguir en caso de problemas. Para lograr esto, deben estar identificados los roles de las personas que interactúan con el área, a fin de que cada uno sepa qué hacer ante la aparición de problemas.

10.1.10 Procedimientos para la verificación de las máquinas de los usuarios

Este procedimiento permitirá encontrar programas que no deberían estar en las máquinas de los usuarios y que, por su carácter, pueden traer problemas de licencias y fuente potencial de virus. El procedimiento debe

explicitar los métodos que se van a utilizar para la verificación, las acciones ante los desvíos y quién/quiénes lo llevarán a cabo.

10.1.11 Procedimientos para el monitoreo de los puertos en la red

Este procedimiento permite saber qué puertos están habilitados en la red, y, en algunos casos, chequear la seguridad de los mismos. El procedimiento deberá describir qué programas se deben ejecutar, con qué reglas, quién estará a cargo de llevarlo a cabo y qué hacer ante las desviaciones detectadas.

10.1.12 Procedimientos de cómo dar a publicidad las nuevas normas de seguridad

Este tipo de procedimiento no siempre es tenido en cuenta. Sin embargo, en una organización es muy importante conocer las últimas modificaciones realizadas a los procedimientos, de tal manera que nadie pueda poner cómo excusa "que no conocía las modificaciones".

En él, debe describirse la forma de realizar la publicidad de las modificaciones: puede ser mediante un mailing, por exposición en transparencias, por notificación expresa, etc.; quién estará a cargo de la tarea y las atribuciones que tiene.

Es fundamental tener en cuenta este último punto ya que un porcentaje de los problemas de seguridad, según está demostrado en estudios de

mercado, proviene del desconocimiento de las normas y/o modificaciones a ellas por parte de los usuarios.

10.1.13 Procedimientos para la determinación de identificación de usuario y grupo de pertenencia por defecto

Este procedimiento determina la forma de establecer las identificaciones y los grupos a los que pertenecerán los usuarios por defecto en el momento de darlos de alta. En él deben explicarse, concisamente, los pasos a seguir para cambiar los derechos y las identificaciones de los usuarios dados de alta y la manera de documentar los mismos, así también como quién será responsable de la tarea.

10.1.14 Procedimientos para recuperar información

Este procedimiento sirve para reconstruir todo el sistema o parte de él, a partir de las copias de seguridad. En él, deben explicarse todos los pasos a seguir para rearmar el sistema a partir de los back-up existentes, así como cada cuánto tiempo habría que llevarlos a cabo y quiénes son los responsables de dicha tarea.

10.1.15 Check-Lists

Las check-lists, como su nombre lo indica, son listas con un conjunto de ítems referentes a lo que habría que chequear en el funcionamiento del sistema. Algunos ejemplos de check-lists:

- Asegurar el entorno. ¿Qué es necesario proteger? ¿Cuáles son los riesgos?
- Determinar prioridades para la seguridad y el uso de los recursos.
- Crear planes avanzados sobre qué hacer en una emergencia.
- Trabajar para educar a los usuarios del sistema sobre las necesidades y las ventajas de la buena seguridad
- Estar atentos a los incidentes inusuales y comportamientos extraños.
- Asegurarse de que cada persona utilice su propia cuenta.
- ¿Están las copias de seguridad bien resguardadas?
- No almacenar las copias de seguridad en el mismo sitio donde se las realiza
- ¿Los permisos básicos son de sólo lectura?

Si se realizan copias de seguridad de directorios/archivos críticos, usar chequeo de comparación para detectar modificaciones no autorizadas. Periódicamente rever todo los archivos de booteo de los sistemas y los archivos de configuración para detectar modificaciones y/o cambios en ellos.

- Tener sensores de humo y fuego en el cuarto de computadoras.
- Tener medios de extinción de fuego adecuados en el cuarto de computadoras.

- Entrenar a los usuarios sobre qué hacer cuando se disparan las alarmas.
- Instalar y limpiar regularmente filtros de aire en el cuarto de computadoras.
- Instalar UPS, filtros de línea, protectores gaseosos al menos en el cuarto de computadoras.
- Tener planes de recuperación de desastres.
- Considerar usar fibras ópticas como medio de transporte de información en la red.
- Nunca usar teclas de función programables en una terminal para almacenar información de login o password.
- Considerar realizar autolog de cuentas de usuario.
- Concientizar a los usuarios de pulsar la tecla ESCAPE antes de ingresar su login y su password, a fin de prevenir los "Caballos de Troya".
- Considerar la generación automática de password.
- Asegurarse de que cada cuenta tenga un password.
- No crear cuentas por defecto o "guest" para alguien que está temporariamente en la organización.
- No permitir que una sola cuenta esté compartida por un grupo de gente.
- Deshabilitar las cuentas de personas que se encuentren fuera de la organización por largo tiempo.

- Deshabilitar las cuentas "dormidas" por mucho tiempo.
- Deshabilitar o resguardar físicamente las bocas de conexión de red no usadas.
- Limitar el acceso físico a cables de red, routers, bocas, repetidores y terminadores.
- Los usuarios deben tener diferentes passwords sobre diferentes segmentos de la red.
- Monitorear regularmente la actividad sobre los gateways.

CAPÍTULO 11

11. CONCLUSIONES Y RECOMENDACIONES

Después de haber analizado los temas mas importantes que se deben considerar al momento de elegir un Servidor Web y las herramientas que se necesitan para su correcto uso y funcionamiento, podemos deducir que no es una tarea fácil.

Todos los Sistemas Operativos analizados en el presente trabajo representan opciones viables para la implementación de seguridad en los servidores. Red Hat Linux es un Sistema Operativo que debe considerarse seriamente ya que presenta numerosas ventajas, además de lo económico de su adquisición, las herramientas de seguridad que incluye hacen factible su configuración como servidor Web.

Los Requerimientos de Hardware para la Instalación de Red Hat Linux son otra ventaja en la utilización de este Software ya que demanda pocos recursos para un funcionamiento óptimo. Por tanto los costos de adquisición de Hardware disminuyen considerablemente en relación a otro Sistema Operativo. Aunque debe verificarse la lista de compatibilidad de Hardware previamente a su adquisición.

Las técnicas de protección estudiadas son soluciones eficientes a los problemas de seguridad, ya que son una combinación de Hardware y

Software capaces de detectar, prevenir y atenuar cualquier situación de peligro para el sistema. La decisión sobre su implantación al sistema está en dependencia de las necesidades de la empresa o del grado de seguridad que se desee adquirir.

No debemos olvidar que "Agregar métodos de seguridad no significa necesariamente un aumento en la seguridad".

Para un desempeño óptimo del servidor deben tomarse muy en cuenta las consideraciones técnicas enunciadas ya que proporcionan un incremento en el rendimiento del sistema según las características de éste. Debe darse mucha importancia a la "seguridad física" del sistema ya que si no se analizan los factores físicos que puedan ocurrir todos los esfuerzos por asegurar un sistema con la tecnología más eficiente no van a servir de nada; se debe pensar más allá de las maneras elementales de sobrepasar los métodos de seguridad, no se debe poner énfasis en una sola manera en que el sistema puede ser atacado, así también como los procedimientos establecidos para el correcto mantenimiento del Servidor Web.

Es necesario tener presente que la seguridad de un sistema no sólo está en dependencia de la calidad del software o del hardware que se utiliza, es parte fundamental seguir ciertas recomendaciones que garantizarán la verdadera seguridad de los sistemas.

Desarrollar políticas de seguridad, las mismas que deben estar detalladas considerando lo siguiente:

- Sencilla y no compleja.
- Fácil de mantener
- Debe construirse con un enfoque hacia la minimización del impacto que los cambios tendrán en su sistema y en sus usuarios.
- Promover la libertad a través de la confianza en la integridad del sistema.
- El reconocimiento de la falibilidad en vez de una falsa sensación de seguridad.
- El enfoque debería estar en los problemas reales en vez de en problemas teóricos.

Otro tema importante enlazado un poco a las políticas son las contraseñas seguras.

Ninguna tarea que demande seguridad es fácil todo depende de ser firmes y seguros en nuestras decisiones, muchas veces se confunde la seguridad con la rigidez, punto equivocado, no debemos olvidar la parte humana para que todo marche de la mejor manera.

BIBLIOGRAFÍA

http://200.75.54.18/libro/sistemas_operativos/sistemas_operativos-node12.html

http://www.ilustrados.com/publicaciones/EpyVVZFykltyaFhnhS.php

curso-sobre.berlios.de/curso/ trab/2004/aorte;ja/seguridadt.pdf

http://www.e-revistaportal.com/presario02..htm

http://www.bsa.org/mexico/press/newsreleases/Estudio-America-Latina-18-05-2005.cfm

http://www.argus.com

http://www.netcraft.com

http://cheops.anu.edu.au/~avalon/examples.html#permissions

http://web.mit.edu/kerberos/www/#what_is

http://www.linux.com/howtos/Linux+IPv6-HOWTO/index.shtml

http://dast.nlanr.net/Projects/Netlog/#intro

http://www-arc.com/products.shtml

http://www.fish.com/satan/summary.html

http://www.fish.com/satan/summary.html

http://www.yolinux.com/TUTORIALS/LinuxTutorialInternetSecurity.html

http://neocodesolutions.com/software/neomail/

http://www.ethereal.com/introduction.html

http://www.apache.org/

http://www.netscape.com/

http://www.tcx.se/

http://www.postgresql.com/

http://www.perl.org/

http://www.php3.de/

http://javaboutique.internet.com/

http://www.cybercursos.net

http://www.agapea.com/JSP-cn320p1i.htm

http://www.apache-ssl.org/#What_is_Apache-SSL

http://www.iec.csic.es/criptonomicon/sslvsshttp.html

GLOSARIO DE TÉRMINOS

Datagrama

Uno o más "paquetes" que constituyen un solo mensaje de red; por ejemplo: " acceder a esta página web"

Nombre de dominio

Un método comprensible de referenciar un grupo de máquinas dentro de Internet; por ejemplo: www.certificadodigital.com.ar, www.tlc-sa.com.ar

IDS

Sistema de detección de intrusiones

Dirección IP

La dirección numérica exclusiva asignada a cada máquina dentro de una red, con el formato 10.20.30.40.999

NIDS

Sistema de detección de intrusiones basado en el análisis del tráfico de una red completa

HIDS

Sistema de detección de intrusiones basado en el análisis del tráfico de un Servidor o Host

Paquete

La unidad más pequeña de datos que se transmite por una red. Un componente de un datagrama fragmentado.

Número de puerto

La manera en que se identifican dentro de una máquina los servicios individuales de red.

Algoritmo

Conjunto de reglas claramente definidas para la resolución de una determinada clase de problemas. La escritura de un programa es sencillamente la elaboración de un algoritmo adecuado para la resolución del problema planteado. Un programa de software es la traducción, en lenguaje de programación, de un algoritmo.

Almacenar

Incluir los datos en una memoria, externa o interna a la computadora, adecuada para conservarlos. Sinónimos de este término son escribir, guardar, grabar y salvar.

Antivirus

Programa que busca y eventualmente elimina los virus informáticos que pueden haber infectado un disco rígido o disquete.

Aplicación

Es el problema o conjunto de problemas para los que se diseña una solución mediante computadora. Ejemplos de aplicaciones son los procesadores de texto (procesamiento o tratamiento de la palabra), las bases de datos (organización y procesamiento de datos) y las hojas de cálculo (organización y procesamiento de números). En Windows se emplea este término indistintamente con el de programa.

Archivo

Es un conjunto de datos relacionados de manera lógica, como puede ser el conjunto de los nombres, direcciones y teléfonos de los empleados de una empresa determinada.

Backup

Copia de seguridad. Se hace para prevenir una posible pérdida de información.

Binario

Es un sistema de numeración en el que los dígitos se representan utilizando únicamente dos cifras, 0 y 1. Como adjetivo indica dos opciones alternativas.

Cracker

Individuo con amplios conocimientos informáticos que desprotege/piratea programas o produce daños en sistemas o redes.

DOS

Siglas de Denial of Service (Denegación de Servicios). Es uno de los tipos de ataques que encontramos actualmente.

FTP

File Transfer Protocol. Protocolo de Transferencia de Archivos. Uno de los protocolos de transferencia de ficheros mas usado en Internet.

Hacker

Experto en informática capaz de de entrar en sistemas cuyo acceso es restringido. No necesariamente con malas intenciones.

Host

Anfitrión. Computador conectado a Internet. Computador en general.

HTML

Hyper Text Markup Language. Lenguaje de Marcas de Hipertexto. Lenguaje para elaborar páginas Web actualmente se encuentra en su versión 3. Fue desarrollado en el CERN (Conseil Europeen pour la Recherche Nucleaire.ConsejoEuropeo para la Investigación Nuclear).

HTTPS

URL creada por Netscape Communications Corporation para designar documentos que llegan desde un servidor web seguro. Esta seguridad es dada por el protocolo

SSL (Secure Sockets Layer) basado en la tecnología de encriptación y autentificación desarrollada por la RSA Data Security Inc.

Internet

Conjunto de redes y ruteadores que utilizan el protocolo TCP/IP y que funciona como una sola gran red. Es la red de redes. Nacida como experimento del ministerio de defensa americano.

Intranet

Se llaman así a las redes tipo Internet pero que son de uso interno, por ejemplo, la red corporativa de una empresa que utilizara protocolo TCP/IP y servicios similares como www, IP Internet Protocol , bajo este se agrupan los protocolos de Internet. También se refiere a las direcciones de red Internet.

IP

Protocolo Internet. Es un protocolo de bajo nivel para redes que describe la manera cómo el usuario puede comunicarse con los miembros Internet. Es la misma IP de TCP/IP

Link

Enlace, unión, hiper enlace. Se llama así a las partes de una página web que nos llevan a otra parte de la misma o nos enlaza con otro servidor.

Login

Entrada de identificación, conexión. Igual que logon.

Packet

Paquete Cantidad mínima de datos que se transmite en una red o entre dispositivos. Tiene una estructura y longitud distinta según el protocolo al que pertenezca. También llamado TRAMA.

Root

Raíz. En sistemas de ficheros se refiere al directorio raíz. En Linux se refiere al usuario principal.

Router

Dispositivo conectado a dos o más redes que se encarga únicamente de tareas de comunicaciones

Shell

Es un procedimiento mediante el cual se puede acceder temporalmente al sistema operativo desde el interior de un programa.

En Windows es una ventana de aplicación especial que permite lanzar otras aplicaciones.

Sniffer

Literalmente "Husmeador". Pequeño programa que busca una cadena numérica o de caracteres en los paquetes que atraviesan un nodo con objeto de conseguir alguna información. Normalmente su uso es ilegal.

TCP/IP

Transmission Control Protocol / Internet Protocol. El término describe dos mecanismos de software empleados para posibilitar la múltiple comunicación entre computadoras de manera libre de error. TCP/IP es el lenguaje común de la Internet, el que permite que diferentes tipos de computadoras utilicen la red y comuniquen unas con otras, indiferentemente de la plataforma o sistema operativo que usen.

Telnet

Protocolo y aplicaciones que permiten conexión como terminal remota a una computadora anfitriona, en una localización remota.

URL

Universal Resource Locator. Nombre genérico de la dirección en Internet, Indica al usuario dónde localizar un archivo HTML determinado, en Internet.

WEB Site

Sitio en el www. Conjunto de páginas web que forman una unidad de presentación, como una revista o libro. Un sitio está formado por una colección de páginas Web.

WWW (World Wide Web)

Servidor de información, desarrollado en el CERN (Laboratorio Europeo de Física de Partículas), buscando construir un sistema distribuido hipermedia e hipertexto.

También llamado WEB y W3. Existen gran cantidad de clientes www para diferentes plataformas.