Universidad Loyola de América

UNIVERSIDAD LOYOLA DE AMÉRICA
SISTEMA INCORPORADO-UNAM CLAVE 8911

ULA

"SEGURIDAD EN APLICACIONES WEB"

T E S I S
Q U E COMO R E Q U I S I T O
PARA OBTENER EL GRADO DE:
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A:

ANTONIO JERÓNIMO RAMOS

DIRECTOR DE TESIS:
MARIO GUILLÉN RODRÍGUEZ

CUERNAVACA MOR. **SEPTIEMBRE DE 2009**

DEDICATORIA

A las dos Concepciones que son mi Guía y mi Fortaleza.

Agradecimientos

A mi madre ejemplo de vida, e impulso para ver hacia arriba siempre.

A mi Familia en Cuernavaca, por que me enseñó a amarme y a creer en mí como persona.

A "Villa de los Niños", lugar donde sembraron en mi la confianza de que todo puede lograrse

A mis Amigos, facetas que en su momento, han reflejado mis alegrías, tristezas, necesidades, sueños y logros.

A la Familia Loyola por creer y apoyar mi proyecto personal.

A mi asesor Dr. Mario Guillen Por su comprensión y orientación en la presente tesis.

A mis maestros, el reconocimiento a su capacidad de transmitirme sus conocimientos y por su amistad.

GRACIAS por compartir y apoyar mis sueños. Nos vemos en la próxima.

Contenido

Lista de Figuras

CAPÍTULO 1

INTRODUCCIÓN

Este capítulo contiene la introducción del trabajo de investigación de tesis. El capítulo esta organizado en las siguientes secciones: Introducción, justificación, definición del problema, propuesta de solución y alcances y limitaciones.

1.1 Introducción.

El uso emergente del protocolo TCP/IP en las redes de computadoras ha facilitado su interconexión y ha permitido enlazar a diferentes redes, lo que ha dado origen un sistema global conocido como el *Internet*. La Internet fue creada inicialmente para ayudar a fomentar comunicaciones entre centros de investigación, universidades y agencias del gobierno. En realidad, la Internet ha experimentado un gran avance durante la última década. Hoy, es la red informática más grande del mundo y ha estado duplicándose en tamaño cada año. Esta proporción de crecimiento mayor que cualquiera otra red creada, incluyendo aún la red telefónica conmutada pública (PSTN).

En la actualidad, dado el crecimiento del Internet y la facilidad de incorporar nuevos sitios y nuevos puntos de consulta ha permitido que todo tipo de persona tenga acceso al Internet. Dando origen a situaciones peligrosas y de alto riesgo, tal como uno lo puede encontrar en la sociedad. Se puede decir que existen usuarios bien intencionados y honrados y personas que intencionalmente tratan de romper los sistemas de seguridad del Internet.

Los problemas de seguridad en la Internet han recibido atención pública, y se han documentado historias de ataques maliciosos de alto perfil, a través del Internet contra el gobierno, negocios, y sitios académicos. El primer incidente más significativo fue el Gusano (Worm), lanzado por Robert T. Mauricio Jr. el 2 de noviembre de 1988[1], que paralizó el Internet. El gusano de Internet afecto a miles de usuarios y despertó una gran inconformidad entre la comunidad navegadora. Este evento generó una gran publicidad y permitió tomar conciencia en asuntos de seguridad en la Internet. A partir de este hecho se tomaron medidas emergentes, como en el instituto de ingeniería del software en la Universidad de Carnegie Mellon (CERT) donde se empezaron a realizar estudios sobre seguridad en redes de computadoras. Desde el incidente del gusano de Internet, los reportes de ataques a la red, tal como descifrado de contraseñas, la falsificación de direcciones IP (Internet Protocol), piratería de sesiones, negaciones de servicios han aumentado.

Existen muchos huecos en el Internet y estos son publicados, lo que atrae la atención de la comunidad de Internet; También existen numerosos incidentes que pasan inadvertidos. A fines de 1996, Dan Farmer [2] realizó un estudio de seguridad en aproximadamente 2,200 computadoras del Internet. Los resultados obtenidos son sorprendentes: dos tercios de los sitios Web tuvieron problemas de seguridad serios. Varios sitios Web de compañías grandes y oficinas federales han sido vandalizadas, y esto se ha convertido en una actividad popular para intrusos de Internet casuales. Más recientemente, los ataques con virus de macro y negación de servicio distribuidos (DDoS) han inquietado a la comunidad de Internet considerablemente.

Actualmente, los individuos, organizaciones comerciales, y agencias de gobierno dependen de la Internet para compartir recursos y su información. En realidad, virtualmente todos en los puntos de acceso y los sitios Web del Internet son vulnerables. Varios estudios han mostrado que muchos individuos y compañías se abstienen de unirse a la Internet simplemente debido a intereses de seguridad. Al mismo tiempo, los analistas están advirtiendo a las compañías sobre las desventajas de no estar unidos a la Internet.

En esta situación contradictoria, la mayoría concuerda que la Internet necesita más y mejor seguridad. En un taller realizado por la Internet Architecture Board (IAB) en 1994, la escalabilidad y la seguridad eran nominados como las dos principales áreas para la arquitectura de Internet. Esto no ha cambiado y es probable que no tenga un cambio sucesivo, debido a que es la columna vertebral de la WWW y las aplicaciones Web.

1.2 Justificación

Realizar un prototipo de sitio Web que contenga las siguientes características:

- Firewall [5]
- Un control de acceso en base de datos.
- Un control de acceso al servidor de aplicación Tomcat [6]
- Un producto de autentificación, por ejemplo, el Verisign [7]

La seguridad en el manejo de la información es hoy tan importante que la propuesta que en este trabajo se genera, intenta ser un apoyo en cualquiera de los niveles antes mencionados, pero sobre todo a nivel de empresas que es donde se pueden presentar problemas de valor incalculable en lo económico sin dejar de lado la importancia del tiempo y desinformación que pueden generar agentes externos.

Se requieren aplicaciones seguros donde se comparte información confidencial a nivel empresarial, y donde se realicen negocios por medio del Internet como es el comercio electrónico y el negocio entre empresas

1.2.1 Recursos

Para el desarrollo del proyecto se requiere una PC conectada a Internet, artículos, libros, revistas con respecto a la seguridad en aplicaciones Web, también se requerirá los siguientes productos:

- Lenguaje Java [8]
- Tomcat [6]
- Firewall [5]
- B.D. MySQL [9]
- Verisign [7]

1.3 Definición del problema

Existen una infinidad de problemas de este tipo pero mencionaremos los más comunes, en los que se propondrá una de varias soluciones posibles. Dada la figura 1.1, que muestra una arquitectura típica de los componentes de Internet, se indicará la problemática en cada uno de estos componentes.

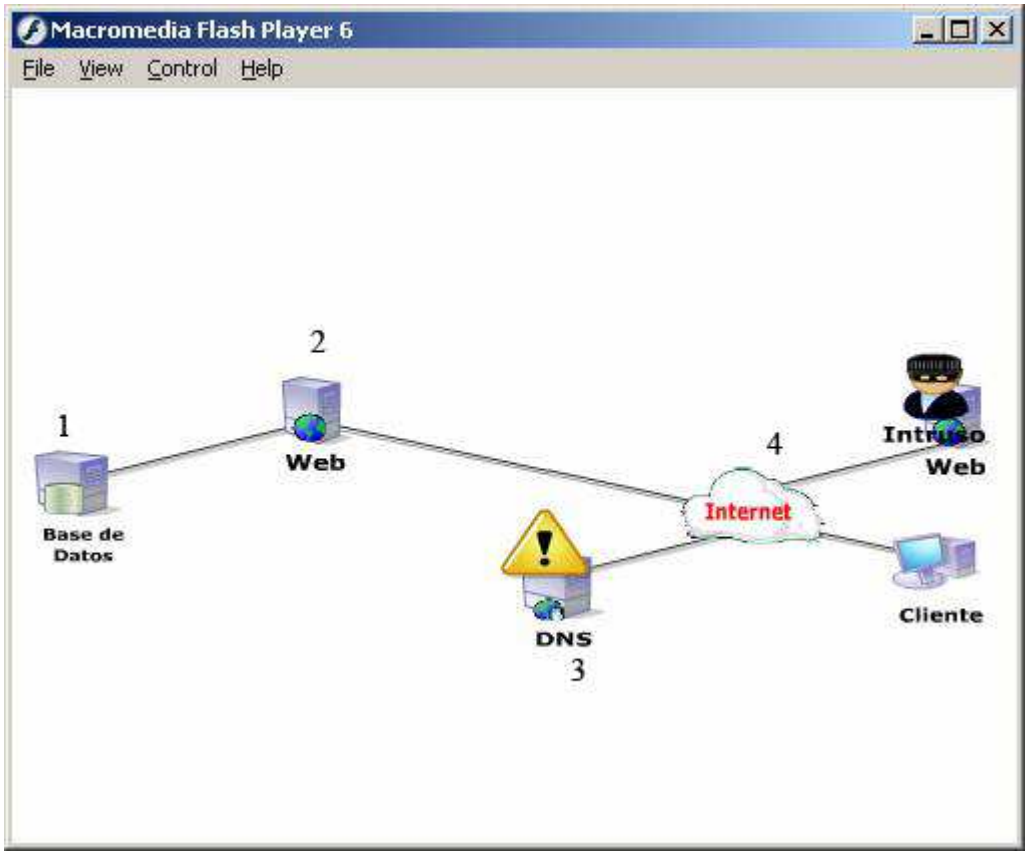


Figura 1.1. Componentes de una aplicación Web Vulnerable

1.3.1 Problemas en el Sistema Administrador de Base de Datos (SABD)

La causa de este tipo de problemas es la falta de verificación del código y colocación de validaciones de los datos en los puntos apropiados. Así como la carencia apropiada de los permisos a los diferentes tipos de usuarios para acceder la base de datos.

1.3.2 Problemas en el Servidor Web (WEB)

La mayoría de los riesgos a nivel informático en las aplicaciones Web se presentan no en la aplicación como tal, sino mas en el servidor Web que las aloja, estas vulnerabilidades son producto de las debilidades de las diferentes implementaciones, no depende solamente del sistema operativo la seguridad, sino de la forma como dicho sistema se relaciona con el servidor de aplicaciones Web; las aplicaciones de tipo Web tienden a ser las más atacadas por su alto grado de exposición, y la facilidad de ataques.

1.3.3 Problemas en el Servidor de Nombres de Dominio

DNS por sus siglas en inglés, *Domain Name Server* consiste en alterar los registros del sistema DNS autorizado para entregar la información a los clientes sobre la ubicación de una URL, en la mayoría de los casos no es una opción difícil de realizar dado que muchos de los DNS implementados están relativamente abiertos a manipulación externa por una pobre seguridad.

1.3.4 Problemas en el Firewall

Muchas veces todos los sistemas son violados sin que el usuario sea enterado y mucho menos saber el daño o la información que ha sido sustraída de su servidor entonces para esto necesitara una pared de fuego que mantenga alejados estos intruso de los servidores.

1.4 Propuesta de Solución.

El objetivo de está propuesta es presentar una metodología de diseño de aplicaciones *Web*, y mostrar por medio de un ejemplo su implementación con las tecnologías adecuadas para el diseño del prototipo de sitio Web que contemple la capacidad de soportar y no permitir la entrada a usuarios no validos no solo a la base de datos sino desde la Web y para esto se utilizaran varios mecanismos de autentificación, y como lo muestra en la figura 2.1 donde están todos los puntos estratégicos de cada red y en donde se especificará el problema, la propuesta y la posible solución de la misma.

1.- SMBD: (Sistema de manejo de base de datos): Se definirán diferentes tipos de usuario de la base de datos y se les asignara diferentes tipos de acceso a las tablas.

2.-WEB: La idea es definir diferentes tipos de usuario para las diferentes aplicaciones Web y que únicamente acceden a esa aplicación.

3.-DNS: (Domain Name Server): Desde el punto de vista del usuario es transparente puesto que para él la aplicación sigue disponible, sin embargo los requerimientos que se hacían en la maquina original ahora se hacen a la maquina del "Intruso" por lo que este sistema es el que puede recolectar toda la información de usuarios y claves, así como los datos del cliente remoto, una vez capturados puede redirigir a los usuarios al sitio falso y el usuario ni el sitio "Web" original nunca se enteran de dicho direccionamiento.

4.-CERTIFICADOS DIGITALES: El proveedor de un servicio de certificación es una entidad de confianza que permite la verificación de identidad de una persona o entidad que quiere utilizar la firma electrónica, o la autenticación de servidores. Da la información sobre la clave pública y otros datos de la persona o entidad incluyéndola los certificados que emita. Debe dar información sobre Uso y Validez de los certificados y ha de ocuparse de mantener actualizada y accesible la lista de revocación de los certificados. Dada la calidad de sus funciones el servicio de certificación ha de ofrecer garantías y demostrar que es una entidad lo suficientemente estable como para que los certificados que emita puedan ser considerados fiables. Ejemplo de ellos son: FESTE, Verisign, IPSCA, ACE, etc. (3).

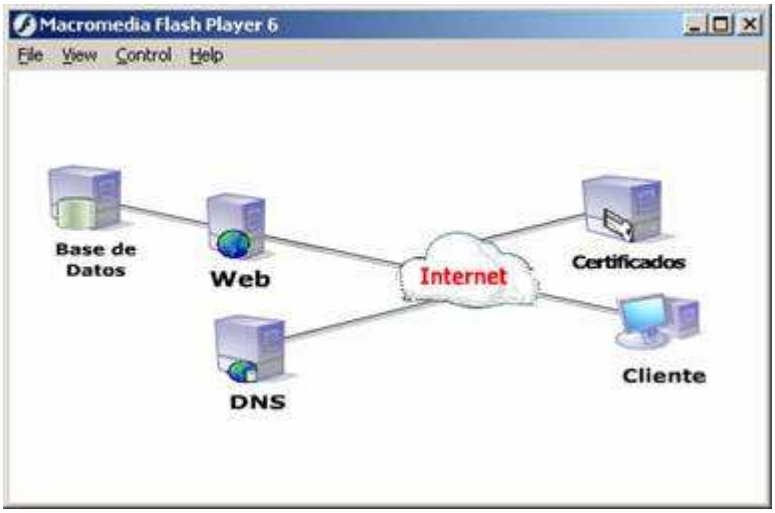


FIG.1.2: Componentes de una aplicación Web activa.

5.-Firewall: El funcionamiento de éste tipo de programas se basa en el "filtrado de paquetes". Todo dato o información que circule entre una PC y la Red es analizado por el programa (firewall) con la misión de permitir o denegar su paso en ambas direcciones (Internet-->PC ó PC---Internet), como lo muestra la figura 1.3, en la que el intruso intenta acceder a la red y es rechazado por la pared de fuego.

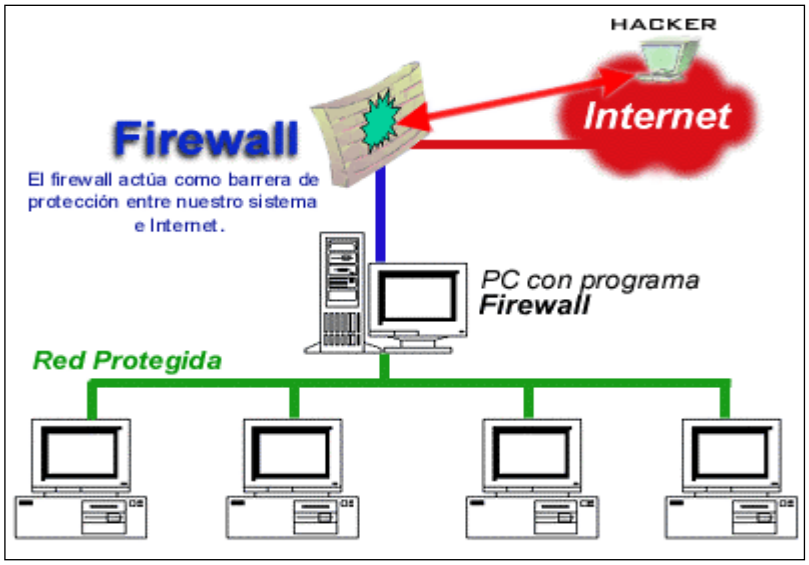


FIG 1.3: Componentes de un Firewall

Se diseñará un prototipo de Aplicación Web, con herramientas de software libre necesarias para diseñar una aplicación Web segura, se hace énfasis de que no se implementarán las herramientas, puesto que también el objetivo es evaluar las herramientas más reconocidas y recomendarle los más eficientes al final de este proyecto. Y se provee que el prototipo resultante sea de utilidad no solo a nivel empresarial o personal sino de una manera educativa y preventiva de los sistemas de seguridad que existen en el mercado y en la red y así usted tomar una decisión correcta ó más acertada con respecto a lo que busca para resguardar su información.

CAPÍTULO 2

ANTECEDENTES

En este capitulo se describen conceptos básicos sobre componentes de una red de aplicaciones en la Web.

2.1 Estado del arte.

2.1.1 Tipos de Seguridad y Autorización en Bases de Datos

La seguridad de las bases de datos es un área amplia que abarca varios temas, entre ellos los siguientes:

- Cuestiones éticas y legales relativas al derecho a tener acceso a cierta información. Es posible que parte de ésta se considere privada y que las personas no autorizadas no puedan tener acceso a ella legalmente.

- Cuestiones de política gubernamental, institucional o corporativa, relacionadas con las clases de información que no deben estar disponibles para el público: por ejemplo, clasificaciones de crédito es historiales médicos personales.

- Cuestiones relacionadas con el sistema, como los niveles del sistema en los que deben realizarse las diversas funciones de seguridad: por ejemplo, si una determinada función de seguridad debería tratarse en el nivel de hardware físico, en el nivel del sistema operativo o en el nivel del SGDB.

- La necesidad en algunas organizaciones de identificar múltiples niveles de seguridad y de clasificar los datos y los usuarios según estos niveles: por ejemplo, máximo secreto (top secret), secreto (secret), confidencial (confidencial), y no confidencial (unclassified). Se debe asegurar el cumplimiento de la política de seguridad de la organización relacionada con el permiso para tener acceso a varias clasificaciones de los datos.

Actualmente se acostumbra a hablar de dos tipos de mecanismos de seguridad de base de datos:

- Los mecanismos de seguridad discrecionales: se usan para conceder privilegios a los usuarios, incluida la capacidad de tener acceso a los archivos de datos, registros o campos específicos en un determinado modo (como lectura, escritura o actualización).
- Los mecanismos de seguridad obligatorios: sirven para imponer seguridad de múltiples niveles clasificando los datos y los usuarios en varias clases (o niveles) de seguridad e implementando después la política de seguridad apropiada de la organización. Por ejemplo, una política de seguridad habitual consiste en permitir a los usuarios con un cierto nivel de clasificación ver solo los elementos de información que están calcificados en el mismo nivel que el usuario (o en un nivel inferior).

Otro problema e seguridad común a todos los sistemas de computador es el de evitar que personas no autorizadas tengas acceso al propio sistema, ya sea para obtener información o para efectuar cambios mal intencionados en una porción de la base de datos el mecanismo de seguridad de un SGBD debe incluir formas de restringir el acceso al sistema como un todo. Esta función se denomina **control de acceso** y se pone en práctica creando cuentas de usuario y contraseñas para que el SGBD controle el acceso de entrada al sistema.

2.1.2 Seguridad en Sistemas Operativos

La protección, es estrictamente un problema interno: ¿Cómo controlamos el acceso a los programas y datos almacenados en un sistema de computación? la seguridad, en cambio, no solo requiere un sistema de protección apropiado, sino también considerar el entorno en el que el sistema opera. La protección interna no es útil si la consola del operador esta al alcance de personal no autorizado, o si los archivos (almacenados, por ejemplo, en cinta y

discos) se pueden sacar simplemente del sistema de computación y llevarse un sistema sin protección. Estos problemas de seguridad son esencialmente de administración, no problemas del sistema operativo.

La información almacenad en el sistema (tanto datos con código), así como los recursos físicos del sistema de computación, tiene que protegerse contra acceso no autorizado, destrucción o alteración mal intencionada, y introducción accidental de inconsistencia.

2.1.2.1 El Problema de Seguridad

Existen varios mecanismos que el sistema operativo puede ofrecer (con ayuda apropiada del hardware) y que permite a los usuarios proteger sus recursos (casi siempre programas y datos). Estos mecanismos funcionan bien en tanto los usuarios no tratan de burlar el uso y acceso debidos a estos recursos. Desafortunadamente, tal situación pocas veces se logra. Cuando no se logra, la seguridad entra en juego. Decimos que un sistema es seguro si sus recursos se usan y acceden como es debido en todas las circunstancias. Por desgracia, en general no es posible lograr una seguridad total. No obstante, debemos contar con mecanismo para hacer que las violaciones de seguridad sean un suceso poco común, en lugar de la norma.

Las violaciones de seguridad del sistema se pueden clasificar como intencionales o accidentales. Es mas fácil protegerse contra un mal uso accidental que contra un abuso mal intencionado. Entre las formas de acceso mal intencionado están:

1. lectura no autorizada de datos (robo de información)
2. modificación no autorizada de datos.
3. destrucción no autorizada de datos

No es posible lograr una protección absoluta del sistema contra un abuso mal intencionad, pero puede hacerse que el costo para el delincuente sea tan alto que frustre la mayor parte de, si no todos, los intentos de acceder, sin la autorización debida, a la información que reside en el sistema.

Para proteger el sistema debemos tomar medidas de seguridad en dos niveles:

- FISICO: el sitio o sitios que contienen los sistemas de computación deben asegurarse físicamente contra el ingreso armado o subrepticio de intrusos.
- HUMANO: los usuarios deben seleccionarse cuidadosamente para reducir la posibilidad de autorizar un usuario que luego dará acceso a un intruso (a cambio de un soborno, por ejemplo).

Se debe mantener la seguridad en ambos niveles para garantizar la seguridad del sistema operativo. Un punto débil en un nivel de seguridad alto (física o humano) permite burlar medidas de seguridad estrictas de bajo nivel (del sistema operativo).
La seguridad dentro del sistema operativo se implementa en varios niveles que van desde contraseñas para acceder al sistema hasta el aislamiento de procesos concurrentes que se ejecutan dentro del sistema. El sistema de archivos también ofrece cierto grado de protección.

2.2 Validación

Un problema de seguridad importante para los sistemas operativos es el de la validación. El sistema de protección depende de una capacidad para identificar los programas y procesos que se están ejecutando. A su vez, dicha capacidad depende en última instancia de que podamos identificar cada usuario del sistema. Un usuario normalmente se identifica a si mismo. ¿Cómo podemos determinar si la identidad de un usuario es autentica? En general, la validación se basa en uno o mas de tres elementos: posesión del usuario (una llave o tarjeta), conocimiento del usuario (un identificador de usuario y una contraseña) y un atributo del usuario (huella dactilar, patrón de retina o firma).

2.2.1 Contraseñas

La estrategia más común para validar la identidad de un usuario es el empleo de contraseñas. Cuando el usuario se identifica con un identificador de usuario o nombre de

cuenta, se le pide una contraseña. Si la contraseña que el usuario proporciona coincide con la que está almacenada en el sistema, éste supone que el usuario esta autorizado.

2.2.1.1 Vulnerabilidades de las contraseñas.

Hay dos formas comunes de adivinara una contraseña. Una es que el intruso (sea humano o un programa) conozca al usuario o tenga información acerca de él. Con demasiada frecuencia, la gente usa información obvia (digamos los nombres de sus mascotas o de sus conyugues) como contraseñas. La otra forma es usar fuerza bruta, probando todas las posibles combinaciones de letras, números y signos de puntuación hasta hallar la contraseña. Las contraseñas cortas no tienen suficientes opciones para impedir que se adivinen mediante intentos repetidos. Por ejemplo, una contraseña de cuatro dígitos decimales solo tienes 10,000 variaciones. En promedio, bastaran 5000 pruebas para lograr un acierto. Si se puede escribir un programa que pruebe una contraseña cada milisegundo, solo tardaría unos cinco segundos en adivinar una contraseña de cuatro dígitos. Las contraseñas mas largas son menos susceptibles a ser adivinadas por enumeración, y los sistemas que distinguen entre letras mayúsculas y minúsculas y que permiten usar números y todos los signos de puntuación en las contraseñas hacen mucho más difícil la tarea de adivinar la contraseña. Desde luego, los usuarios deben aprovechar el espacio de contraseña más grande y no usar, por ejemplo, solo letras minúsculas.

2.2.2 Amenaza por programas

En un entorno en el que un programa escrito por un usuario podría ser utilizado por otro usuario, existe una posibilidad de abuso que podría dar pie a un comportamiento inesperado de los cuales hay dos métodos comunes para causar tal comportamiento: los caballos de Troya y las puertas traseras o secretas.

2.2.2.1 Caballo de Troya

Muchos sistemas cuentan con mecanismo que permite a aun usuario ejecutar programas escritos por otros usuarios. Si tales programas se ejecutan en un dominio que proporciona los derechos de acceso del usuario ejecutante, podrían abusar de esos derechos. Por ejemplo, un programa editor de textos podría contener código para buscar ciertas palabras clave en el archivo que se va a editarse se encuentran, todo el archivo podría copiarse en un área especial accesible para el creador del editor de textos. Un segmento de código que abusa de su entorno se denomina caballo de Troya.

2.2.2.2 Puerta trasera o secreta.

El diseñador de un programa o sistema podría dejar un "agujero" en el software que solo el puede usar. Por ejemplo, el código podría verificar que el identificador de usuario o la contraseña tengan un valor específico, y podría burlar los procedimientos de seguridad normales. Ha habido casos de programadores que han sido arrestados por desfalco de bancos mediante la inclusión de errores de redondeo en su código, haciendo que de vez en cuando medio centavo se abone a sus cuentas. Estos abonos pueden sumar cantidades apreciables de dinero, si consideramos el número de transacciones que un banco grande ejecuta.

Uno de los aspectos más fascinantes y más dinámicos de la seguridad de la red se relaciona con los ataques. La mucha atención de los medios y muchas ofertas del producto del vendedor han estado apuntando ataques y metodologías del ataque. Ésta es quizás la razón que por la que se ha estado enfocando muchas preguntas en esta área en particular.

2.3 Tipos de Ataque

Por lo general los ataques se clasifican en cuatro grupos:

■ **Los ataques activos**: Estos incluyen DOS, negación del servicio distribuida (DDoS), el desbordamiento del almacenador intermediario, ataque síncrono (SYN), *spoofing*, el rapto del Protocolo de control de transmisión del Internet (TCP/IP), wardialing, ataques de replay, el salto del dumpster, la ingeniería social y la exploración de la vulnerabilidad.

■ **Ataques pasivos**: éstos incluye sniffing, y spoofing.

■ **Ataques de password**: éstos incluye fuerza bruta y ataques diccionario-basados de la contraseña

■ **Deficiencia de programación y los ataques criptográficos:** éstos incluyen: backdoors, virus, Trojans, gusanos, rootkits, la explotación del software, botnets y ataques matemáticos.

2.3.1 Ataques activos

Los ataques activos se pueden describir como ataques en los cuales el atacante está procurando activamente causar daño a una red o a un sistema.

Los ataques activos tienden a ser muy visibles, porque el daño causado es a menudo muy sensible. Algunos de los ataques activos más conocidos son:

- DOS/DDoS
- Desbordamientos del almacenador intermediario
- Ataques de SYN

- Spoofing del Internet Protocol (IP); éstos y muchos más se detallan a continuación.

2.3.1.1 Ataque DOS

Este tipo de ataque no implica romper el sistema del blanco. El objetivo fundamental de un ataque del DOS es degradar el servicio, si es recibido por un solo servidor o entregado por una red entera infraestructura. Un ataque del DOS procura reducir la capacidad de un sitio a los clientes del servicio, si esos clientes son usuarios físicos o entidades lógicas tales como otros sistemas informáticos. Esto se puede alcanzar por cualquiera que sobrecarga la capacidad del apuntar la red o el servidor para manejar tráfico entrante, o enviando los paquetes de la red que hacen sistemas y redes de blanco comportarse imprevisible. Desafortunadamente para el administrador, el comportamiento "imprevisible" traduce generalmente a un sistema colgado o estrellado. Aunque los ataques del DOS por la definición no generan un riesgo a confidencial o los datos sensibles, pueden actuar como herramienta eficaz para enmascarar actividades más intrusas que podrían ocurrir simultáneamente.

2.3.1.2Ataques de DDoS

Algunas formas de ataques DOS se pueden amplificar por los intermediarios múltiples, el primer paso de una hazaña del DOS se origina de una sola máquina. Sin embargo, los ataques de DOS se han desarrollado más allá de una sola escala (inundación de SYN) y de ataques de dos niveles (del smurf). Los ataques de DDoS avanzan un paso adelante más doloroso del enigma uno del DOS. Las metodologías modernas del ataque ahora han abrazado el mundo de computar de varias filas distribuido. Una de las diferencias significativas en la metodología de un DDoS el ataque es que consiste en dos fases distintas. Durante la primera fase, el perpetrador compromete las computadoras dispersadas a través del Internet y les instala un software especializado en estos anfitriones para la ayuda en el ataque. En la segunda fase, los anfitriones comprometidos (designados zombis)

a través de los intermediarios mandan (llamados master) para comenzar el ataque. El modelo de varias filas de ataque DDoS y a su capacidad a los paquetes del spoof y cifrar comunicaciones, puede hacer seguir abajo del delincuente verdadero un proceso tortuoso. La estructura del comando que apoya un ataque de DDoS puede ser absolutamente enrollada (véase el Figura 2.1), y él puede ser difícil de determinar una terminología que lo describa claramente. Miremos a una de las convenciones de nombramiento más comprensibles para una estructura del ataque de DDoS y los componentes implicados.

Los componentes de software implicados en un ataque de DDoS incluyen:

- Cliente del el software de control usado por el hacker para lanzar ataques. El cliente dirige secuencias de comando a sus anfitriones subordinados.

- El software del *demon* del programa el funcionamiento en un zombi que recibe entrante secuencias y acto de comando del cliente en ellos por consiguiente. El demonio es proceso responsable realmente de poner el ataque en ejecución detallado en las secuencias de comando.

Los anfitriones implicados en un ataque de DDoS incluyen:

Master: una computadora que corra el software cliente
Zombi: Es un anfitrión subordinado que corre el proceso del *daemon*.
Blanco: El objetivo del ataque

En los anfitriones no señalados como zombis, el hacker instala el software (llamado un demonio) usado para enviar corrientes del ataque. El demonio funciona en el fondo en el zombi, esperando un mensaje para activar el software de la hazaña y para lanzar un ataque apuntado en la víctima señalada. Un *daemon* puede lanzar t múltiples tipos de ataques,

tales como User Datagram Protocol (UDP) o inundaciones de SYN. Combinado con la capacidad de utilizar spoofing, el *daemon* puede demostrar ser una herramienta muy flexible y de gran alcance del ataque.

Para reclutar los anfitriones para el ataque, la blanco de los hackers son las maquinas con poca seguridad conectados al Internet. Los Hackers utilizan varias técnicas de inspección ya sea automáticos y manuales para descubrir las redes y los anfitriones asegurados inadecuadamente. Después de que se hayan identificado las máquinas inseguras, el atacante compromete los sistemas de varias maneras. La primera tarea que un hacker cuidadoso es borrar evidencia que comprometa a su sistema, y también asegurarse de que el anfitrión comprometido aprobará una reexaminación precipitada. Algunos de los anfitriones comprometidos se convierten en *master*, mientras que otros son utilizados como zombis. Los *masters* reciben las órdenes que entonces dejan filtrar a través a los zombis de quienes son responsables. El amo es solamente responsable de enviar y de recibir mensajes cortos del control, haciendo redes más bajas de la anchura de banda.

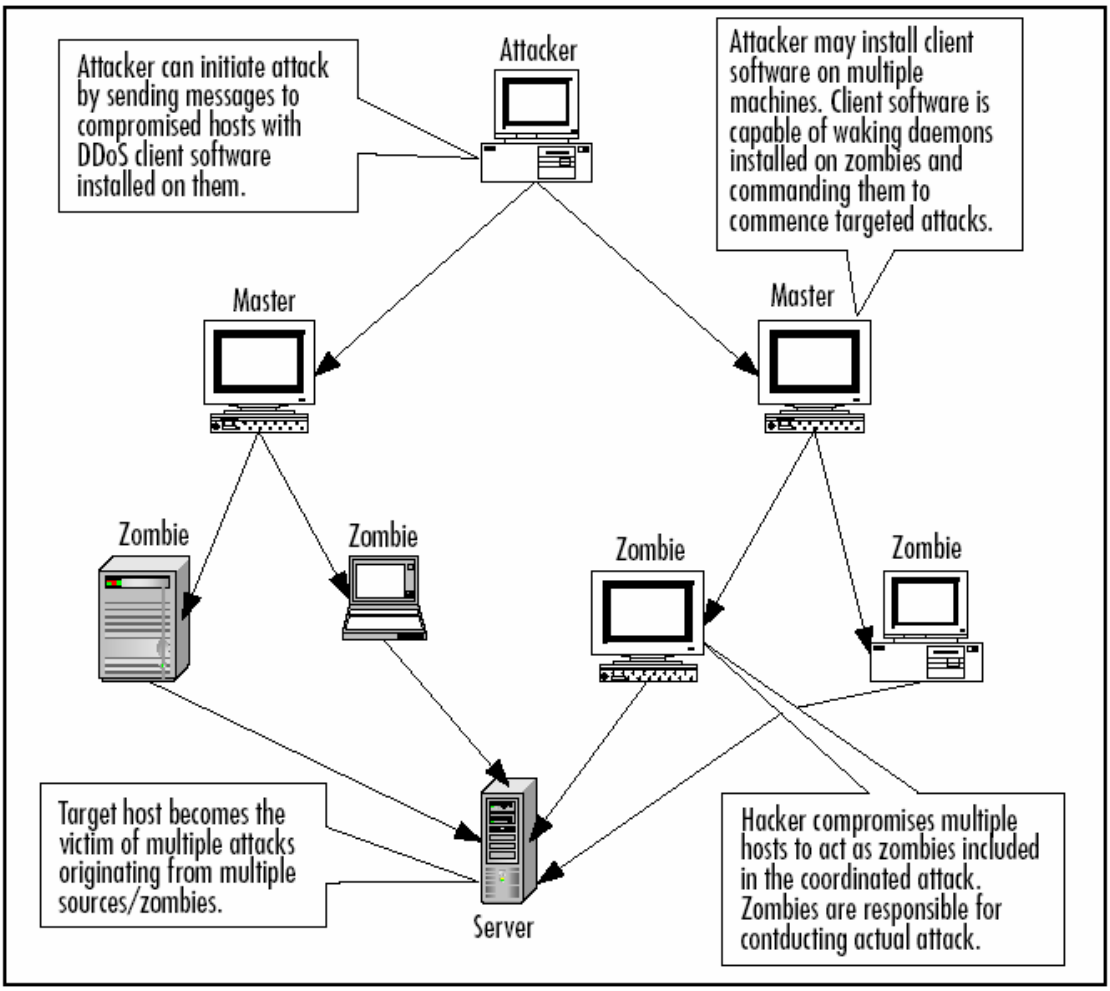


FIG. 2.1 diagrama de ataques DDos

2.3.1.3 Ataques de SYN

SYN es un bit de control dentro del segmento TCP, que se utiliza para sincronizar los números de secuencia iniciales ISN de una conexión en el procedimiento de establecimiento de tres fases (*3 way handshake*)

Un ataque de SYN es un ataque del DOS que explota una debilidad básica encontrada en el TCP/IP el protocolo, y su concepto es bastante simple. El comportamiento previsto es que el anfitrión que inicia envía un paquete de SYN, a el cual el anfitrión que responde publicará un SYN/ACK y esperará una contestación del ACK del iniciador. Con un ataque de SYN, o la inundación de SYN, el atacante envía simplemente solamente el SYN

paquete, saliendo de la víctima que espera una contestación. El ataque ocurre cuando el atacante envía millares y millares de paquetes de SYN a la víctima, forzándolos esperar las contestaciones que nunca vienen. Mientras que el anfitrión está esperando la contestación, no puede aceptar cualquier petición legítima, así que llega a ser inaccesible, así alcanzando el propósito de un ataque del DOS como lo muestra la figura 2.1

Algunos cortafuegos protegen contra ataques de SYN reajustando hasta que finalicen conexiones después de un descanso específico. Otra protección está con el uso de las cookies de SYN, donde una computadora bajo ataque responde con un paquete especial de SYN/ACK y no espera una respuesta del ACK. Solamente cuando el paquete del ACK en respuesta al paquete de SYN/ACK vuelve, hace la entrada generan una entrada de la coleta de la información dentro del paquete especial de SYN/ACK.

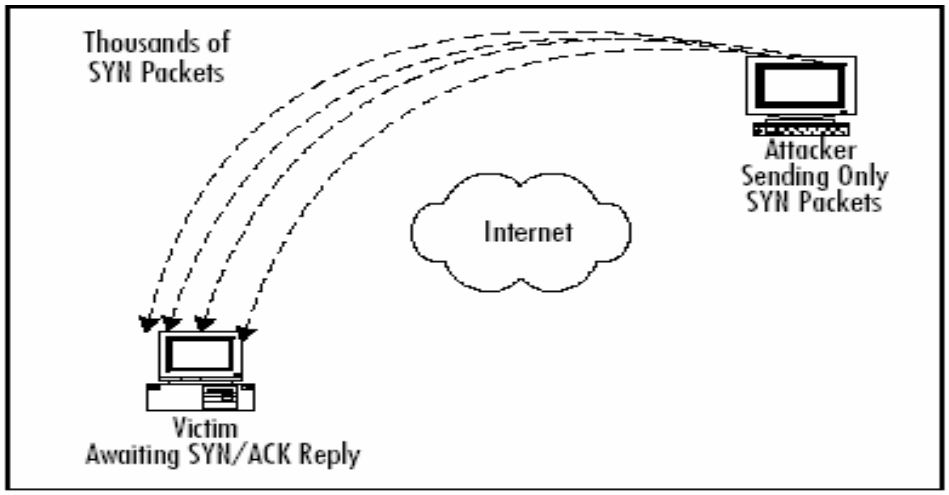


Fig. 2.2 Diagrama de ataque SYN

2.3.1.4 *Hacking* del TCP/IP

El TCP/IP que secuestra, o la sesión que secuestra, es un problema que tiene aparecido adentro la mayoría Usos de TCP/IP, extendiéndose de sesiones simples del telnet a los Web-basados de uso comercial. Para secuestrar una conexión del TCP/IP, un usuario malévolo primero inserta los datos de un usuario legítimo, y en seguida se inserta en la

sesión como un ataque de MITM. Una herramienta conocida es el HUNT (www.packetstormsecurity.org/sniffers/hunt/) es de uso muy general para supervisar sesiones del secuestro. Trabaja especialmente en sesiones básicas del telnet o del File Transfer Protocol (FTP). Una forma más interesante y más malévola de sesión que secuestra implica Web-basado usos (especialmente e-comercio y otros usos que confían en el encendido de *cookies* para mantener el estado de la sesión). El primer panorama implica el secuestro de la *cookies* de un usuario, que se utiliza normalmente para almacenar las credenciales de la conexión y la otra información sensible, y el usar de esa *cookies* entonces para tener acceso a la sesión de ese usuario. El usuario legítimo recibe un mensaje de que simplemente su "sesión expiró" o mensaje fallado "conexión" e incluso no estará probablemente enterado de que algo sospechoso sucede. La otra edición con Web los usos del servidor que pueden conducir a la sesión que secuestra son suspensiones incorrectamente configurados de la sesión. Un uso del Web se configura típicamente a la suspensión de la sesión de un usuario después de un período de inactividad en el sistema. Si esta suspensión es demasiado grande, deja una ventana de oportunidad para que un atacante potencialmente utilice una *cookies* secuestrada o aún prediga un número de la identificación de la sesión y secuestre la sesión de un usuario. Para prevenir estos tipos de ataques, como con otros ataques de TCP/IP - basado, el uso de sesiones cifradas es clave; en el caso de usos del Web, las identificaciones únicas y pseudo-random de la sesión y las *cookies* se deben utilizarse asegurando los puntos de la capa de cifrado (SSL). Esto hace que los atacantes les sea difícil insertarse en las conexiones, o para interceptar las comunicaciones como lo muestra la figura 2.3.

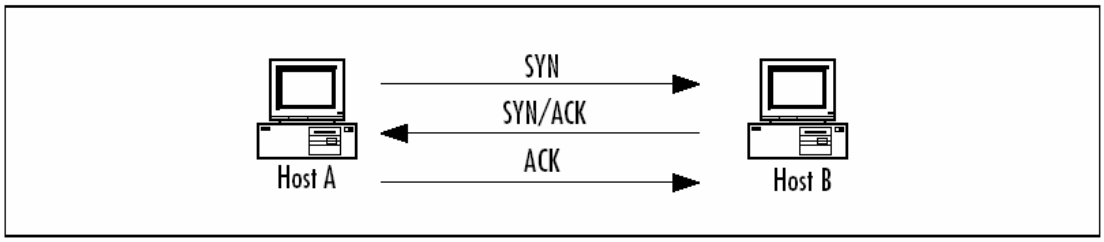


Fig. 2.3: Estándar del TCP/IP

2.3.1.5 Ataques de *Spoofing*

Spoofing significa el abastecimiento de la información falsa sobre tu identidad para tener el acceso desautorizado a los sistemas. Estos ataques pueden ser IP, E-mail, o Web site *spoofing*. IP *Spoofing* El ejemplo más clásico de *spoofing* es IP spoofing.TCP/IP requiere que cada anfitrión complete su propia dirección de la fuente en los paquetes, y no hay casi medidas parar alertar a los anfitriones de la mentira. *Spoofing*, por la definición, es siempre intencional. Sin embargo, el hecho de que algunos malfuncionamientos y desconfiguraciones pueden causar exactamente el mismo efecto que un *spoof* intencional, aunque es difícil determinar si una dirección indica un *spoof*.

2.3.1.5.1 E-mail Spoofing

El Spam es un problema importante en Internet de hoy. Y algunas de las técnicas que los *spammers* utilizan incluyen e E-mail spoofing, donde el remitente del E-mail cambia el campo del E-mail de modo que aparezca que el mensaje vino de e-mail confiable en fuente o dominio. Pocos usuarios abrirían un E-mail de defcon@xploits.com que contiene un archivo "Screensaver.scr atractivo," pero los usuarios abrirían mucho más fácil un archivo llamado las "vacaciones Schedules.xls" de hr@yourcompany.com.

Un E-mail spoofing es extremadamente fácil de hacer, y difícil de parar. La educación del usuario es la mejor defensa contra el E-mail spoofing, junto con la configuración apropiada de los programas de protección del E-mail que la compañía tiene.

2.3.1.5.2 Web site Spoofing

El Web site spoofing ocurre cuando un atacante crea un sitio Web muy similar, si no idéntico, a otro sitio, generalmente a un e-comercio, a actividades bancarias, o a una destinación de juego. El propósito principal del sitio Web spoofing es trampear a los visitantes en que piensan que ellos está utilizando el sitio original, así que entrarán en sus

credenciales (username, contraseña, PIN, y así sucesivamente), que serán capturadas por los dueños del sitio spoofed.

2.3.1.6 *Phishing*

Phishing es una combinación del E-mail y del Web site spoofing, y es uno de los ataques más peligrosos actualmente activos. El ataque básico de Phishing comienza con spammer. Enviando correos masivos que personificaban un sitio Web que tienen *spoofed*. Una vez que el usuario vaya a el sitio *spoofed*, es muy difícil distinguirlo del verdadero. Una vez que el usuario entre con sus credenciales, lo vuelven a redireccionar generalmente al sitio verdadero después de que el atacante tenga robado las credenciales, y el usuario no tiene ninguna idea de qué sucedió. La mejor manera de protegerse contra phishing y el Web site spoofing es guardar siempre tu browser del Internet remendado, y comprobar la barra de la dirección de URL para verificar el sitio correcto está conectado. Phishing es tan peligroso que las versiones más últimas del comandos en los buscadores, incluyendo Internet Explorer, Firefox y ópera, han agregado built-in protección phishing, y sitios como Google y eBay ofertan su propia barra de herramientas para incluir la protección anti-phishing.

2.3.1.7 Salto del Dumpster

El salto del Dumpster es el proceso de cavar a través de la basura de una víctima en un intento por ganar la información. Es a menudo fácil encontrar la información del cliente o de producto, las notas internas, y la información uniforme de la contraseña que se han puesto en papelera de reciclaje. En un ejemplo famoso, una compañía importante de la ropa había desechado simplemente fotos e información sobre su formación próxima de la ropa. No duró para la información negligentemente desechada al viento para arriba en las manos de competidores, hacer gran daño a los planes de la compañía de la víctima para un lanzamiento de producto único.

2.3.1.8 Ingeniería Social

La ingeniería social se pasa por alto a menudo en planes y panoramas de la seguridad, que es desafortunado, porque es uno de los métodos más peligrosos y fácilmente más usados para infiltrar la red de una víctima. El concepto que parece mentira y más que creativa; es el juego de un artista. Las mentiras son sostenidas a menudo por los materiales encontrados en *dumpste*r al zambullirse, que implica cavar a través de la basura de la víctima, el buscar importantes documentos, listas del teléfono, y así sucesivamente. Una manera mucho más fácil de conseguir la información sobre una víctima potencial es el Web site de la compañía, que enumera generalmente al personal ejecutivo, las listas de sus teléfonos, es la otra información que se puede utilizar para trampear a una víctima. Saber los pocos nombres importantes, por ejemplo, pueden hacer que el atacante parezca más auténtico y puede permitir pedir la información clasificada sobre el teléfono. Esta información puede ser algo tan trivial como algún número de teléfono, o tan confidencial como la identificación de la contraseña de alguien y de la conexión del servidor Desafortunadamente, no se puede emplear un cortafuego para restringir este uso, porque el accesa con las claves de la persona correcta que obtuvo especialmente por el teléfono o vía el email. El factor humano puede a menudo ser el acoplamiento más débil de la seguridad de una red. Sin embargo, el lado positivo es que la mayoría de los empleados no desean dañar la compañía, y seguirán los procedimientos del acceso si son enterados del problema. Es muy importante reconocer la amenaza que la ingeniería social plantea. La educación del empleado y crear una política de protección de contraseña son las mejores maneras a defender contra la ingeniería social.

2.3.2 Ataques pasivos

Durante un ataque pasivo, el contrario del ataque activo que es directo, en el ataque pasivo el atacante no está afectando directamente la red de la víctima. El atacante está esperando a escuchar que algo ocurrir, o está intentando recopilar la información. Algunos ataques pasivos se pueden comparar a escuchar detrás de la puerta la conversación de alguien, o espiar a alguien. Hay bastantes maneras interesantes que los ataques pasivos pueden ocurrir, tales como.

2.3.2.1 Sniffing y Eavesdropping

Un sniffer es una herramienta que permite a una máquina ver todos los paquetes que están pasando a través (o aéreo en una red inalámbrica), incluso los que no están destinados para ese anfitrión. Esta es una técnica de gran alcance para diagnosticar problemas de la red, pero también puede ser utilizada maliciosamente para explorar las contraseñas, el E-mail, o cualquier otro tipo de datos enviados. Para husmear una función, la tarjeta de la red tiene que ser configurada en el modo promiscuo (que permite que procese todos los paquetes en el alambre). Después de capturar una cantidad de datos, un atacante puede volver a montar fácilmente las páginas Web vistas, los archivos descargados, o el E-mail enviado, todo con el tecleo del ratón.

Eevesdropping se fia del uso de *keyloggers*. Éstos son los programas que funcionan ocultado en el OS, y registran todas las llaves mecanografiadas por el usuario. Contraseña, las cuentas, *usernames*, y se pueden descubrir más que con un *keylogger* que funciona en una máquina sin sospecha. Algunos *keyloggers* incluso toman intervalos regulares de la pantalla y los envían al dueño del programa (o al atacante). Para protegerse contra *keyloggers*, se deben activar regularmente programas contra-virus y del anti-spyware.

2.3.3Ataque de password

2.3.3.1 Ataques de contraseña o por fuerza bruta

La fuerza bruta, en su definición más simple, refiere a intentar tantas combinaciones de la contraseña como sea posible hasta encontrar el correcto. Es un método de uso general para obtener las contraseñas, especialmente si la lista cifrada de la contraseña está disponible. El número de las combinaciones posibles de la contraseña es finito y es por lo tanto vulnerable al ataque de la fuerza bruta.

2.3.3.2 Ataques Diccionario-basados

La selección apropiada de la contraseña reducir al mínimo las contraseñas triviales aun asi no esta totalmente protegido y esta latente la posibilidad de que su contraseña pueda ser violada. Las contraseñas simples tales como cualquier palabra individual en una lengua hacen las contraseñas más débiles porque pueden ser violadas con un ataque elemental del diccionario. En este tipo de ataque, las listas largas de palabras de una lengua particular llamando los archivos de diccionario para que busquen una similitud con las contraseña cifrada. Contraseñas más complejas que incluyen letras, los números, y los símbolos requieren una técnica diversa de la fuerza bruta que incluya todos los caracteres imprimibles y toma mas tiempo encontrar la contraseña entonces el atacante deja de intentarlo.

2.3.3.3 Ataques malévolos del código

Los ataques del código son programas cuidadosamente hechos a mano escritos por los atacantes y diseñados para hacer daño. Los caballos de Troya, virus, spyware, rootkits, y malware, son todos los ejemplos de esta clase de ataque. Estos programas se escriben para ser independientes y no requerir siempre la intervención del usuario o para el atacante estar presente para tu daño que se hará.

2.3.4 Ataques por deficiencia de programación y criptográficos

2.3.4.1 Virus

Un virus de la computadora se define como programa de computadora que repliega un programa para interfiera con un hardware, el software o el OS. Un virus se diseña para replegar y para eludir la detección. Como cualquier otro programa de computadora, un virus debe ser ejecutado por una función en especifico (debe ser cargada en la memoria de computadora) y entonces la computadora debe seguir las instrucciones del virus. Esas instrucciones constituyen la carga útil del virus. La carga útil puede interrumpir o cambiar ficheros de datos, exhibir un mensaje, o hacer el OS funcionar incorrectamente. Otros virus existentes tienen la capacidad de unirse a los programas de otra manera legítimos. Esto podría ocurrir cuando se crean los programas abiertos, o aún mas modificados, los gusanos

2.3.4.2 Worm (gusano)

Es un programa de uno auto-repliegue que no altera archivos sino reside en memoria activa y se duplica por medio de las redes de ordenadores. Los gusanos funcionan automáticamente dentro de OS y de software y son invisibles al usuario. A menudo, no se notan en sistemas hasta que los recursos de la red se consumen totalmente, o el funcionamiento de la PC de la víctima se degrada a los niveles inutilizables. Algunos gusanos no sólo se auto-repliegan sino que también contienen una carga útil malévola. Hay muchas maneras de las cuales los gusanos pueden ser transmitidos, incluyendo E-mail, Cuartos de la charla de Internet (Chat`s), programas del P2P, y por supuesto el Internet.

2.3.4.3 Caballos de Troya

Un caballo de Troya se asemeja de cerca a un virus, pero está realmente tiene su propia categoría. El caballo de Troya se refiere a menudo como la forma más elemental de código malévolo. Un caballo de Troya se utiliza de manera semejante como esta narrado en la Iliada de Homero; es un programa dentro del cual el código malévolo. Se disfraza lo más a menudo como algo divertido, tal como a juego nuevo. Se oculta el programa malévolo, y cuando este se activa para realizar su función, puede arruinar realmente tu disco duro.

2.3.4.4 El rootkit

A este tipo de malware que intenta encubrir su presencia del OS y de los programas del antivirus en una computadora. Su nombre viene del mundo de UNIX, donde los hackers intentan establecerse en el directorio raíz de una computadora después de que la infectan. Un rootkit puede modificar los bloques básicos de un OS como los conductores del núcleo o de la comunicación, o sustituir los programas del sistema de uso general por versiones del rootkit.

2.3.4.5 Una puerta trasera (backdoors)

Es esencialmente cualquier programa o configuración deliberada diseñada a tener en cuenta el acceso alejado a un sistema. Troyanos, los rootkits, e incluso los programas legítimos se pueden utilizar para instalar una puerta trasera.

2.3.4.6 Bombas lógicas

Una bomba lógica es un tipo de malware que se puede comparar a una bomba de tiempo. Son diseñados para hacer daño después de cumplir cierta condición. Éste puede ser al pasar de una cierta fecha u hora. Una bomba lógica bien conocida fue el virus de Chernobyl se activó hasta cierta fecha, en este caso, el 26 de abril, el aniversario del desastre de Chernobyl. En ese día, el virus causó estragos procurando reescribir el BIOS del sistema de la víctima y borrando el disco duro.

2.4 2.4 Como Evitar ataques pasivos y activos.

2.4.1 Evitar ataques DoS/DDoS

Un DDoS es una ampliación del ataque DoS, se efectúa con la instalación de varios agentes remotos en muchas computadoras que pueden estar localizadas en diferentes puntos del mundo. El atacante consigue coordinar esos agentes para así, de forma masiva, amplificar el volumen de saturación de información (*flood),* pudiendo darse casos de un ataque de cientos o millares de computadoras dirigidos a una máquina o red objetivo. Esta técnica se ha revelado como una de las más eficaces y sencillas a la hora de colapsar servidores, la tecnología distribuida a amplia y más sofisticada hasta el punto de otorgar el poder de causar daños serios a una persona con escasos conocimientos técnicos.

La falla reportada por DNS Report dice así:

«ERROR: *One or more of your nameservers reports that it is an open DNS server. This usually means that anyone in the world can query it for domains it is not authoritative for (it is possible that the DNS server advertises that it does recursive lookups when it does not, but that shouldn't happen). This can cause an excessive load on your DNS server. Alos, it is strongly discouraged to have a DNS server be both authoritative for your domain and be recursive (even if it is not open), due to the potential for cache poisoning (with no*

recursion, there is no cache, and it is impossible to poison it). Alos, the bad guys could use your DNS server as part of an attack, by forging their IP address»

Significa que el servidor DNS puede permitir a cualquiera realizar consultas recursivas. Si se trata de un DNS que se desea pueda ser consultado por cualquiera, como puede ser el caso del DNS de un ISP, esto es normal y esperado. Si se trata de un servidor que solo debe consultar la red local, o bien que se utiliza para propagar dominios hospedados localmente, si es conveniente tomar medidas al respecto.

Para dar solución al problema: ir al fichero /etc./named.conf se añade en la sección de opciones (options) una línea que defina la red, las redes o bien los ACL que tendrán permitido realizar todo tipo de consultas.

```
options {
        directory "/var/named";
        dump-file "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        forwarders { 192.168.0.1; };
        forward first;
        allow-recursion { 127.0.0.1; 192.168.0.0/24; };
};
```

Lo anterior hace que solo 192.168.0.0/24 pueda realizar todo tipo de consultas en el DNS, ya sea para un nombre de dominio hospedado localmente y otros dominios resueltos en otros servidores (ejemplo: www.yahoo.com, www.google.com, www.linuxparatodos.net, etc). El resto del mundo solo podrá realizar consultas sobre los dominios hospedados localmente y que estén configurados para permitirlo.

```
options {
        directory "/var/named";
        dump-file "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
```

```
        forwarders { 192.168.0.1; };
        forward first;
        allow-recursion { 127.0.0.1; 192.168.0.0/24; };
};

zone "." IN {
        type hint;
        file "named.ca";
};

zone "miredlocal" {
        type master;
        file "miredlocal.zone";
        allow-update { none; };
        allow-query { 192.168.0.0/24; };
        allow-transfer { 192.168.0.2; };
};

zone "midominio.com" {
        type master;
        file "midominio.com.zone";
        allow-update { none; };
        allow-transfer { 200.76.185.252; 200.76.185.251; };
};
```

Una configuración como la anterior hace lo siguiente:

Red Local: cualquier tipo de consulta hacia dominios externos y locales (es decir, www.yahoo.com, www.google.com, linuxparatodos.net, además de midominio.com).
Resto del mundo: solo puede hacer consultas para la zona de midominio.com
De este modo se impide que haya consultas recursivas y con esto impedir la posibilidad de sufrir/participar de un ataque DDoS

2.4.2 Evitar ataques SYN

¿Qué amenazas ayuda a reducir?

Para reducir el impacto en un host que sufre un ataque SYN, TCP/IP minimiza la cantidad de recursos dedicados a conexiones TCP incompletas y reduce el tiempo antes de abandonar la conexión. Cuando se detecta un ataque SYN, TCP/IP en Windows Server 2003 y Windows XP reduce el número de retransmisiones del segmento SYN-ACK y no asigna memoria ni recursos de entradas de la tabla a la conexión hasta que se haya completado la negociación de protocolos tridireccionales de TCP.

Puede controlar la protección ante ataques SYN mediante la configuración del Registro SynAttackProtect en HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters (tipo REG_DWORD). Establezca SynAttackProtect en **0** para deshabilitar la protección ante ataques SYN y en **1** para habilitarla.

Para TCP/IP en Windows XP (todas las versiones) y Windows Server 2003 sin ningún Service Pack instalado, SynAttackProtect se establece en **0** de forma predeterminada. Para TCP/IP en Windows Server 2003 con SP1, SynAttackProtect se establece en **1** de forma predeterminada.

2.4.2.1 API auxiliares para IP de notificación de ataques SYN

Para que una aplicación notifique a los administradores de la red que se está produciendo un ataque SYN, la API auxiliar para IP permite el uso de las API de notificación de ataques SYN llamadas NotifySecurityHealthChange y CancelSecurityHealthChangeNotify. Aún no se ha publicado información acerca de estas nuevas API en Microsoft Developer Network (MSDN).

2.4.2.2 Asignación inteligente de puertos TCP

TCP/IP en el SP1 de Windows Server 2003 ha implementado un algoritmo de asignación inteligente de puertos TCP. Cuando una aplicación solicita un puerto TCP disponible, TCP/IP intenta primero encontrar un puerto disponible que no corresponda a una conexión en estado de intervalo de espera. Si no encuentra ninguno, escoge cualquier puerto disponible.

2.4.3 Evitar Ataques de Spoofing

Para evitar ataques de spoofing exitosos contra nuestros sistemas podemos tomar diferentes medidas preventivas; en primer lugar, parece evidente que una gran ayuda es reforzar la secuencia de predicción de números de secuencia TCP: un esquema de generación robusto puede ser el basado en [Bel96], que la mayoría de Unices son capaces de implantar (aunque muchos de ellos no lo hagan por defecto). Otra medida sencilla es eliminar las relaciones de confianza basadas en la dirección IP o el nombre de las máquinas, sustituyéndolas por relaciones basadas en claves criptográficas; el cifrado y el filtrado de las conexiones que pueden aceptar nuestras máquinas también son unas medidas de seguridad importantes de cara a evitar el spoofing.

2.4.4 Evitar ataques *Phishing*

Entre las acciones que pueden llevarse a cabo para no ser víctima del "phishing" destacan las siguientes:

- Desconfiar de cualquier mensaje que solicite datos confidenciales como nombres de usuario, contraseñas, números de tarjeta de crédito, etc.

- Para asegurarnos de que conectamos con un servidor Web seguro comprobar, en la barra de direcciones del navegador, que la URL comienza por "https://"

- Prestar atención al icono -como, por ejemplo, un candado cerrado en el caso de Internet Explorer-, que debe aparecer en la barra de estado inferior del navegador, informando de

que conectamos con un servidor Web seguro. Además, haciendo doble click sobre el icono podrá visualizarse el certificado de seguridad y comprobar su validez.

- No utilizar enlaces para acceder a sitios Web con información confidencial, y menos aún si proceden de mensajes de correo electrónicos o páginas no fiables. En su lugar, se recomienda escribir en el navegador la dirección correspondiente.

- Mantener el sistema puntualmente actualizado, tanto el sistema operativo, como el resto de aplicaciones, especialmente el navegador que se utilice para las transacciones electrónicas.

- Contar con una solución antivirus actualizada, ya que también proliferan los gusanos, troyanos y *keyloggers* (o programas que capturan las pulsaciones de teclado) destinados a robar los datos de los usuarios para poder acceder a la banca electrónica y a otros sitios con información confidencial.

2.4.5 Programas Spyware y Adware Spyware

Como su nombre implica, espían las máquinas donde están instalados. Recopilan la información personal, con o sin el permiso del usuario, y la utilizan para muchos propósitos. Spyware se ha convertido en un problema tan penetrante que se han creado programas anti-spyware. Hay muchos de tipos de spyware en términos de su propósito, su método de instalación, sus métodos de colección, y así sucesivamente. Los propósitos pueden incluir la comercialización (demostrando anuncios mientras que hojea, también llamado adware), cambio de dirección del tráfico (lleva a usuarios a los sitios que no se propusieron visitar), e incluso propósitos criminales (robando las contraseñas y los números de la tarjeta de crédito, enviándolo al creador de los spyware). Spyware se puede instalar por los usuarios que los descargan de sitios Web, pero a menudo los trampean en la

instalación del spyware, instalado secretamente como parte de la instalación de otra utilidad, o utilizar una vulnerabilidad explotable en *browsers*.

CAPÍTULO 3

PROTOTIPO

En este capitulo se mostrara lo estructura y los componente del prototipo de seguridad en aplicaciones Web en la red.

3.1 Prototipo

3.1.1 Descripción del prototipo de seguridad en aplicación Web (AW)

En esta sección se presentan el prototipo de seguridad en AW la interacción externa e interna de los módulos que lo componen, así como el flujo de la información y el flujo de control.

3.1.2 Representación del contenido de la información (Prototipo de seguridad en AW).

En la figura 3.1 se presenta el modelo a nivel externo, el cual se basa en una arquitectura de 3 capas, se incluyen los usuarios (puntos de acceso), la red externa (Internet) y los componentes del sitio Web.

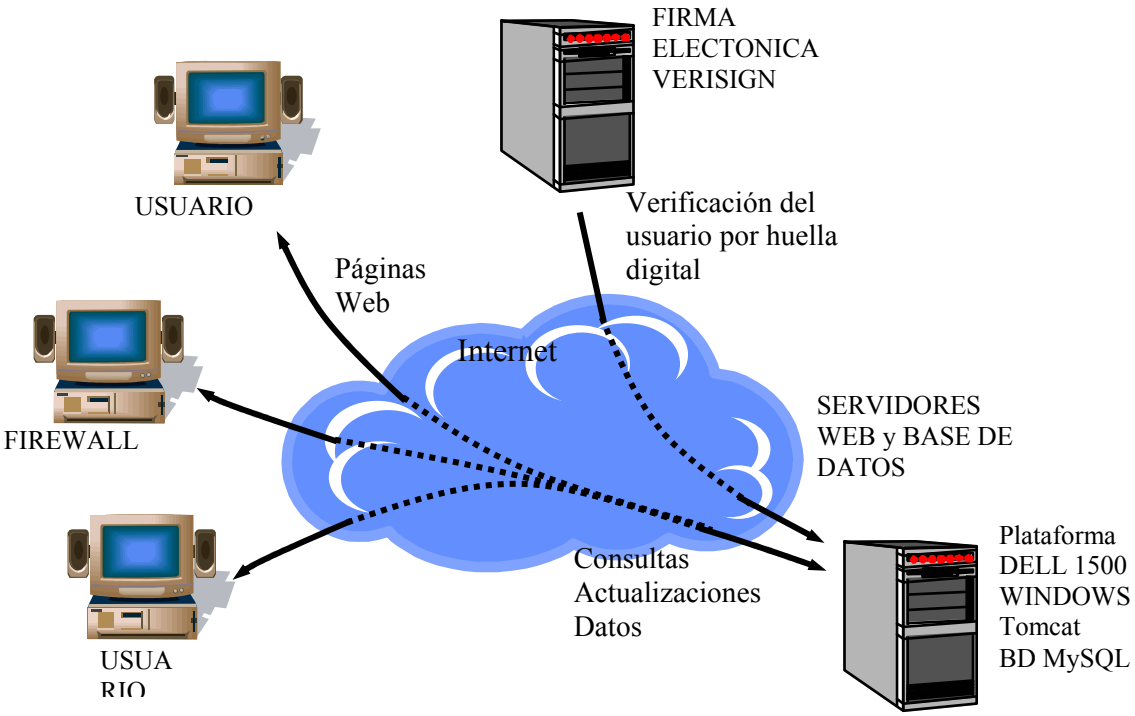


Fig. 3.1 Interacción del PROTOTIPO DE AW.

El sistema está formado por dos servidores: uno como servidor de la base de datos, y el segundo como autentificador del usuario en acceso a la base de datos ambos ubicados en la misma plataforma de cómputo.

Los usuarios pueden interactuar con el prototipo de AW por medio de un visualizador Web comercial, accediendo al sitio Web a través de la Internet. La información presentada en las páginas Web será obtenida por el servidor de BD a través de la interacción con los usuarios, solo accederán a la base de datos a través del servidor Web.

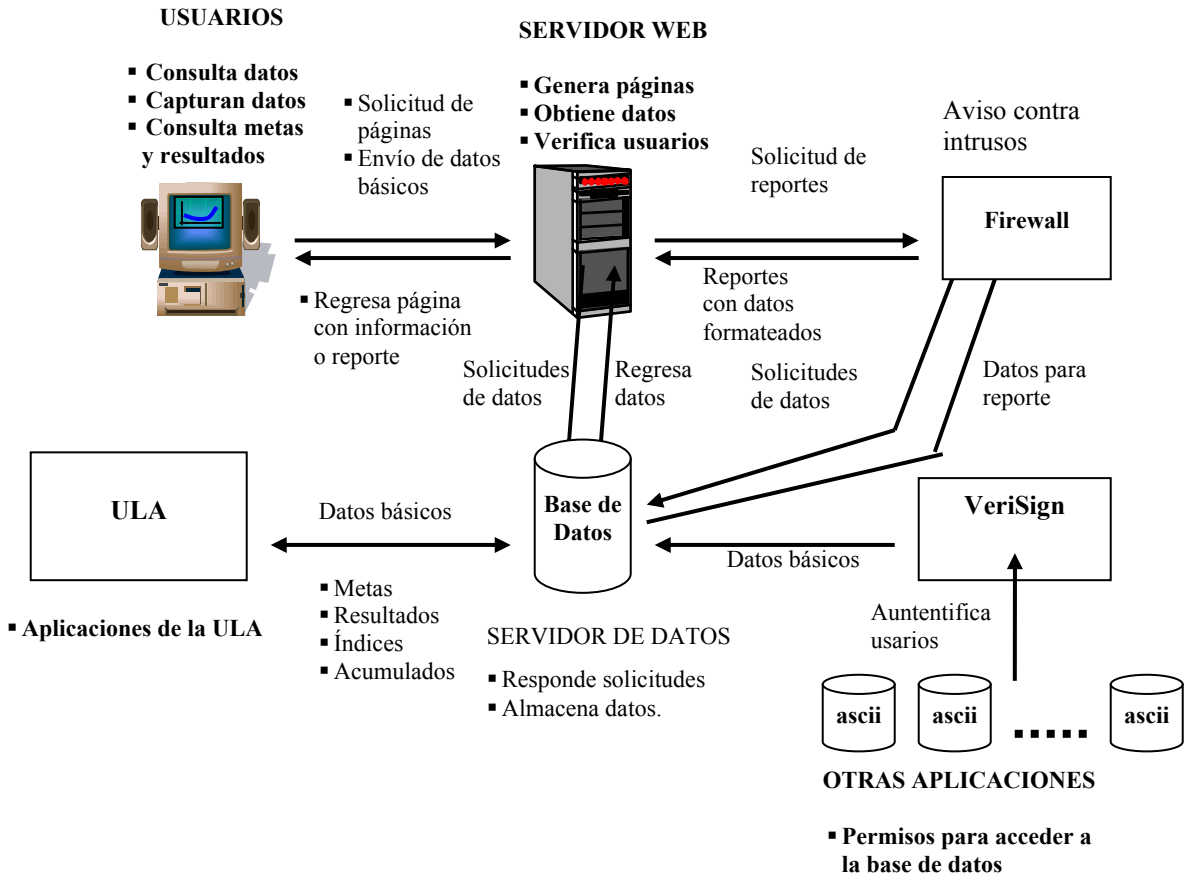
El sistema cuenta con un módulo para la administración del sistema y los permisos de acceso.

La plataforma del servidor Web y del servidor de base de datos está integrada por un equipo Dell 1500, con el sistema operativo Windows xp, el servidor Apache (Tomcat) y el manejador de base de datos MySQL.

3.2Representación del flujo de la información.

3.2.1 Flujo de datos.

En la figura 3.2 se muestra el diagrama del flujo de datos del sistema.



3.3 Componentes de sistema

Existen una infinidad de problemas de este tipo pero mencionaremos los más comunes, en los que se propondrá una de varias soluciones posibles. La figura 3.3, muestra los componentes típicos de Internet, se indicará la problemática en cada uno de estos componentes.

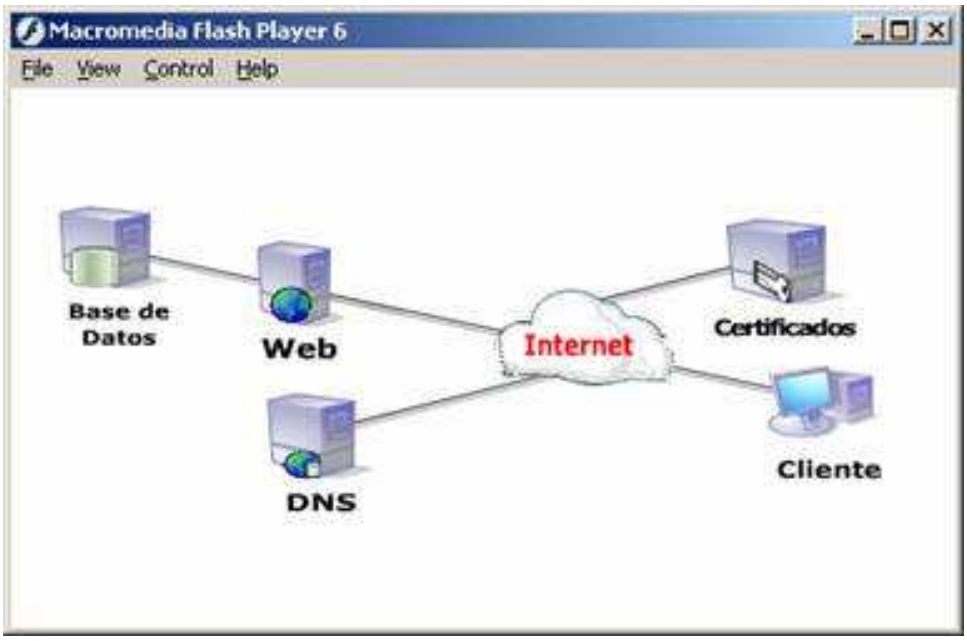


Fig. 3.3: Componentes del Sistema.

3.3.1 Sistema Administrador de Base de Datos (SABD)

La causa de este tipo de problemas es la falta de verificación del código y colocación de validaciones de los datos en los puntos apropiados. Así como la carencia apropiada de los permisos a los diferentes tipos de usuarios para acceder la base de datos.

3.3.1.1 Modelos de bases de datos

Un modelo de datos es básicamente una "descripción" de algo conocido como *contenedor de datos* (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de *base de datos*; por lo general se refieren a algoritmos, y conceptos matemáticos.

3.3.2 Los modelos más utilizados en las bases de datos son:

3.3.2.1 Bases de datos jerárquicas

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se conocen como *hojas*. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

3.3.2.2 Bases de datos de red

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de *nodo*: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

3.3.2.3 Bases de datos relacionales

3.3.2.3.1 Modelo relacional

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, [10] de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por *registros* (las filas de una tabla), que representarían las tuplas, y *campos* (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o *Lenguaje Estructurado de Consultas*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

3.3.2.4 Bases de datos orientadas a objetos

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia - Propiedad a través de la cual los objetos heredan atributos y comportamiento dentro de una jerarquía de clases.
- Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

Se está trabajando en SQL3, que es el estándar de SQL92 ampliado, que soportará los nuevos conceptos orientados a objetos y mantendría compatibilidad con SQL92.

3.3.2 Servidor Web (WEB)

La mayoría de los riesgos a nivel informático en las aplicaciones Web se presentan no en la aplicación como tal, sino mas en el servidor Web que las aloja, estas vulnerabilidades son producto de las debilidades de las diferentes implementaciones, no depende solamente del sistema operativo la seguridad, sino de la forma como dicho sistema se relaciona con el servidor de aplicaciones Web; las aplicaciones de tipo Web tienden a ser las más atacadas por su alto grado de exposición, y la facilidad de ataques. Un servidor Web es un programa que implementa el *protocolo HTTP* (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un formato de archivo y HTTP es un protocolo.

Cabe destacar el hecho de que la palabra *servidor* identifica tanto al programa como a la máquina en la que dicho programa se ejecuta. Existe, por tanto, cierta ambigüedad en el término, aunque no será difícil diferenciar a cuál de los dos nos referimos en cada caso.

Un servidor Web se encarga de mantenerse a la espera de *peticiones HTTP* llevada a cabo por un *cliente HTTP* que solemos conocer como *navegador*. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear *www.google.com* en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Sobre el servicio Web *clásico* podemos disponer de aplicaciones Web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java(applets), flash, Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas *scripts*). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje *javascript* y *java*, aunque pueden añadirse mas lenguajes mediante el uso de *plugin*s
- Aplicaciones en el lado del servidor: el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones Web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad adicional, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador Web básico puede utilizar este tipo de aplicaciones. Las aplicaciones Web en el lado del servidor ser pueden programar en algunos de los siguientes lenguajes:

- *PHP*
- *ASP*
- *Perl*
- *CGI*
- JAVA
- JSP (Tecnología Java)

Algunos servidores Web importantes son:

- Tomcat
- *Apache*
- *IIS*

Otros servidores, más simples son:

- lighttpd
- thttpd

Existen servidores Web que implementan el protocolo http seguro (https); el sistema HTTPS utiliza un cifrado basado en las Secure Socket Layers (SSL) para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. Cabe mencionar que el uso del protocolo HTTPS no impide que se pueda utilizar HTTP. Es aquí,

cuando nuestro navegador nos advertirá sobre la carga de elementos no seguros (HTTP), estando conectados a un entorno seguro (HTTPS).

Es utilizado principalmente por entidades bancarias, tiendas en línea, y cualquier tipo de servicio que requiera el envío de datos personales o contraseñas.

3.3.3 Servidor de Nombres de Dominio

(DNS, por sus siglas en inglés, Domain Name Server): DNS es un sistema jerárquico con estructura de árbol. El inicio se escribe "." y se denomina raíz, al igual que en las estructuras de datos en árbol. Bajo la raíz se hallan los dominios de más alto nivel (TLD, del inglés, Top Level Domain), cuyos ejemplos más representativos son ORG, COM, EDU y NET, si bien hay muchos más. Del mismo modo que un árbol, tiene una raíz y ramas que de ella crecen. Si el lector está versado en ciencias de la computación, reconocerá el DNS como un árbol de búsqueda y será capaz de encontrar en él los nodos, nodos hoja y otros conceptos.

Cuando se busca una máquina, la consulta se ejecuta recursivamente en la jerarquía, empezando por la raíz. Si se desea encontrar la dirección IP de ftp.akane.linuxsilo.net., el servidor de nombres (del inglés, nameserver) tiene que empezar a preguntar en algún sitio. Empieza mirando en su caché. Si conoce la respuesta, pues la había buscado anteriormente y guardado en dicha caché, contestará directamente. Si no la sabe, entonces eliminará partes del nombre, empezando por la izquierda, comprobando si sabe algo de akane.linuxsilo.net., luego de linuxsilo.net., luego net. y, finalmente, de ".", del cual siempre se tiene información ya que se encuentra en uno de los ficheros de configuración en el disco duro. A continuación preguntará al servidor "." acerca de ftp.akane.linuxsilo.net. Dicho servidor "." no sabrá la contestación, pero ayudará a nuestro servidor en su búsqueda dándole una referencia de dónde seguir buscando. Estas referencias llevarán a nuestro servidor hasta el servidor de nombres que conoce la respuesta.

Así pues, empezando en "." encontramos los sucesivos servidores de nombres para cada nivel en el nombre de dominio por referencia. Por supuesto, nuestro servidor de nombres guardará toda la información obtenida a lo largo del proceso, a fin de no tener que preguntar de nuevo durante un buen rato.

En el árbol análogo, cada "." en el nombre es un salto a otra rama. Y cada parte entre los "." son los nombres de los nodos particulares en el árbol. Se trepa el árbol tomando el nombre que queremos (ftp.akane.linuxsilo.net) preguntando a la raíz (".") o al servidor que sea padre desde la raíz hacia ftp.akane.linuxsilo.net acerca de los cuales tengamos información en la caché. Una vez se alcanzan los límites de la caché, se resuelve recursivamente preguntando a los servidores, persiguiendo las referencias (ramas) hacia el nombre.

Otro concepto del cual no se habla tanto, pero que no es menos importante, es el dominio in-addr.arpa, que también se encuentra anidado como los dominios "normales". in-addr.arpa nos permite hacernos con el nombre del host cuando tenemos su dirección. Merece la pena destacar aquí que las direcciones IP están escritas en orden inverso en el dominio in-addr.arpa. Si se tiene la dirección de una máquina tal como 192.168.0.1, el servidor de nombres procederá del mismo modo que con el ejemplo ftp.akane.linuxsilo.net. Es decir, buscará los servidores arpa., luego los servidores in-addr.arpa., luego los 192.in-addr.arpa., luego los 168.192.in-addr.arpa. y, por último, los servidores 0.168.192.in-addr.arpa. En este último encontrará el registro buscado: 1.0.168.192.in-addr.arpa. de ahí que se pueda llegar a manipular que consiste en alterar los registros del sistema DNS autorizado para entregar la información a los clientes sobre la ubicación de una URL, en la mayoría de los casos no es una opción difícil de realizar dado que muchos de los DNS implementados están relativamente abiertos a manipulación externa por una pobre seguridad.

3.3.4 Firewall

Muchas veces todos los sistemas son violados sin que el usuario sea enterado y mucho menos saber el daño o la información que ha sido sustraída de su servidor entonces para evitar esto se necesitara una pared de fuego que mantenga alejados estos intruso de los servidores. Como su nombre lo indica, un firewall es una pared de fuego debido a la semejanza práctica con la técnica usada para combatir incendios forestales, donde se crean líneas controladas de fuego en el terreno para impedir que las llamas avancen más allá de las mismas.

En una red, por lo general, se trata de un programa destinado a impedir el tráfico indiscriminado de información entre la red de la empresa e Internet, con el fin de evitar intrusiones o ataques a la información.

La manera en la que operan los firewalls es relativamente simple. El administrador de la red puede indicar qué tipo de información puede ser admitida, así como los datos que pueden cruzar hacia el exterior.

Mucha gente ignora que una computadora capaz de navegar por Internet es también accesible desde la Red. El problema real se suscita cuando el equipo forma parte de una red y contiene información privada y, en algunos casos, de gran importancia para la empresa.

Un hacker experimentado puede encontrar huecos de seguridad en un equipo a través de Internet y virtualmente entrar a la computadora en cuestión, lo cual no sólo implica que podría ver o manipular los archivos que se encuentren almacenados en ésta, sino también tomar control parcial del equipo e incluso ejecutar aplicaciones en él.

El firewall ofrece distintos servicios como filtrado de paquetes y servidor proxy. En el primer caso, el programa se encarga de revisar cada paquete de datos desde y hacia Internet y sólo aquellos que logren pasar a través de los filtros determinados por el administrador de la red podrán llegar a su destino.

Es posible indicar que ciertas computadoras de la red (por medio de su dirección IP) no puedan emitir o recibir datos fuera de la misma, también se puede establecer que no sea aceptada información proveniente de ciertas direcciones, que se filtren paquetes con información específica o se deshabiliten ciertos protocolos de comunicación por Internet.

3.3.4.1 Tipos de Firewall

3.3.4.1.1 Firewall de capa de red

Funciona al nivel de la red de la pila de protocolos (TCP/IP) como filtro de paquetes IP o bien a nivel 2, de enlace de datos, no permitiendo que estos pasen el cortafuegos a menos que se atengan a las reglas definidas por el administrador del cortafuegos o aplicadas por defecto como en algunos sistemas inflexibles de cortafuegos. Una disposición más permisiva podría permitir que cualquier paquete pase el filtro mientras que no cumpla con ninguna regla negativa de rechazo.

3.3.4.1.2 Firewall de capa de aplicación

Trabaja en el nivel de aplicación. Analizan todo el tráfico de HTTP, (u otro protocolo), puede interceptar todos los paquetes que llegan o salen desde y hacia las aplicaciones que corren en la red. Este tipo de cortafuegos usa ese conocimiento sobre la información transferida para proveer un bloqueo más selectivo y para permitir que ciertas aplicaciones autorizadas funcionen adecuadamente. A menudo tienen la capacidad de modificar la información transferida sobre la marcha, de modo de engañar a las aplicaciones y hacerles creer que el cortafuegos no existe. Otros también tienen incorporan software adicional para realizar un filtrado más pormenorizado del tráfico a nivel de aplicación, como puede ser un software antivirus para tráfico http o smtp así como incluir sistemas de detección de intrusos.

3.3.4.2Ventajas de un Firewall

- Protección selectiva.- Solamente entran a la red las personas autorizadas basadas en la política de la red en base a las configuraciones.
- Optimización de acceso.- Identifica los elementos de la red internos y optimiza que la comunicación entre ellos sea más directa. Esto ayuda a reconfigurar los parámetros de seguridad.
- Protección de información privada.- Permite el acceso solamente a quien tenga privilegios a la información de cierta área o sector de la red.
- Protección contra virus.- Evita que la red se vea infestada por nuevos virus que sean liberados.

Un dispositivo de Proxy (que realiza una acción por otro) puede actuar como cortafuego respondiendo a los paquetes de entrada como si fuera una aplicación mientras que bloquea otros paquetes.

Los cortafuegos tienen a menudo funcionalidad de traducción de direcciones de red (NAT) y es común utilizar el así llamado espacio de direcciones privado en las máquinas detrás de ella. Este espacio de direcciones privado se realiza cuando no se dispone de suficientes direcciones públicas de Internet. El espacio de direcciones privado está definido por el RFC 1918.

3.3.4.3 Ejemplo de implantación de un Firewall

La imagen muestra el funcionamiento de un firewall de capa de red, permitiendo el libre acceso de la red interna hacia otros dispositivos y al exterior mientras que el firewall restringe el acceso a ciertos servicios y usuarios de Internet.

Un *router* puede incluir firewall integrado así como permite la creación de Redes Privadas Virtuales, ayudando a optimizar los recursos al mismo tiempo un firewall puede también ser configurado en una simple PC aplicado de la misma manera, a través de iptables y otras funciones de sistemas operativos como GNU/Linux o BSD.

Cabe agregar que la mayoría, sino todos, poseen una zona desmilitarizada (DMZ), que permite dejar sin efecto en ella el firewall y exponer, por tanto, una porción de la red protegida.

La idea principal de un cortafuegos es crear un punto único de control de acceso para la entrada y salida de tráfico de una red. Un cortafuegos adecuadamente configurado es sistema bastante efectivo de añadir protección a una instalación informática, pero en ningún caso debe considerarse que poseer una única línea de defensa es suficiente. La seguridad en profundidad, en múltiples capas, es la metodología óptima para lograr una red más segura ante ataques tanto internos como externos.

3.3.4.4 Algunos cortafuegos comerciales

Estos son algunos de los cortafuegos comerciales disponibles en el mercado.

- Enterasys Networks Dragon IDS (Enterprise/Empresarial)
- Outpost Firewall Pro
- Sygate Personal Firewall Pro
- Tiny Firewall 2008 Pro
- Webroot desktop Firewall
- Private Firewall
- Kaspersky Anti-Hacker
- Kerio Personal Firewall
- Deelfield Personal Firewall
- Conseal PC Firewall
- Norton Personal Firewall
- Panda Internet Security
- McAfee Personal Firewall
- Black Ice
- Look 'n' Stop Lite
- eBox Platform (software libre)

CAPÍTULO 4

PROPUESTA DE SOLUCIÓN

En este capitulo se mencionan las propuestas de solución a los problemas vistos con anterioridad.

4.1 Propuesta de solución

El objetivo de está propuesta es presentar una metodología de diseño de aplicaciones *Web*, y mostrar por medio de un ejemplo su implementación con las tecnologías adecuadas para el diseño del prototipo de sitio Web que contemple la capacidad de soportar y no permitir la entrada a usuarios no validos no solo a la base de datos sino desde la Web y para esto se utilizaran varios mecanismos de autentificación, y como lo muestra en la siguiente figura están todos los puntos estratégicos de cada red en donde se especificará el problema, la propuesta y la posible solución de la misma.

4.1.1 SMBD (Sistema de manejo de base de datos)

Se definirán diferentes tipos de usuario de la base de datos y se les asignara diferentes permisos de acceso a las tablas.

4.1.2WEB

La idea es definir diferentes tipos de usuario para las diferentes aplicaciones Web y que únicamente acceden a esa aplicación.

4.1.3 DNS: (Domain Name Server)

Desde el punto de vista del usuario es transparente puesto que para él la aplicación sigue disponible, sin embargo los requerimientos que se hacían en la maquina original ahora se hacen a la maquina del "Intruso" por lo que este sistema es el que puede recolectar toda la información de usuarios y claves, así como los datos del cliente remoto, una vez capturados puede redirigir a los usuarios al sitio falso y el usuario ni el sitio "Web" original nunca se enteran de dicho direccionamiento.

4.1.4 Certificados digitales

El proveedor de un servicio de certificación es una entidad de confianza que permite la verificación de identidad de una persona o entidad que quiere utilizar la firma electrónica, o la autenticación de servidores. Da la información sobre la clave pública y otros datos de la persona o entidad incluyéndola los certificados que emita. Debe dar información sobre Uso y Validez de los certificados y ha de ocuparse de mantener actualizada y accesible la lista de revocación de los certificados. Dada la calidad de sus funciones el servicio de certificación ha de ofrecer garantías y demostrar que es una entidad lo suficientemente estable como para que los certificados que emita puedan ser considerados fiables. Ejemplo de ellos son: FESTE, Verisign, IPSCA, ACE, etc.

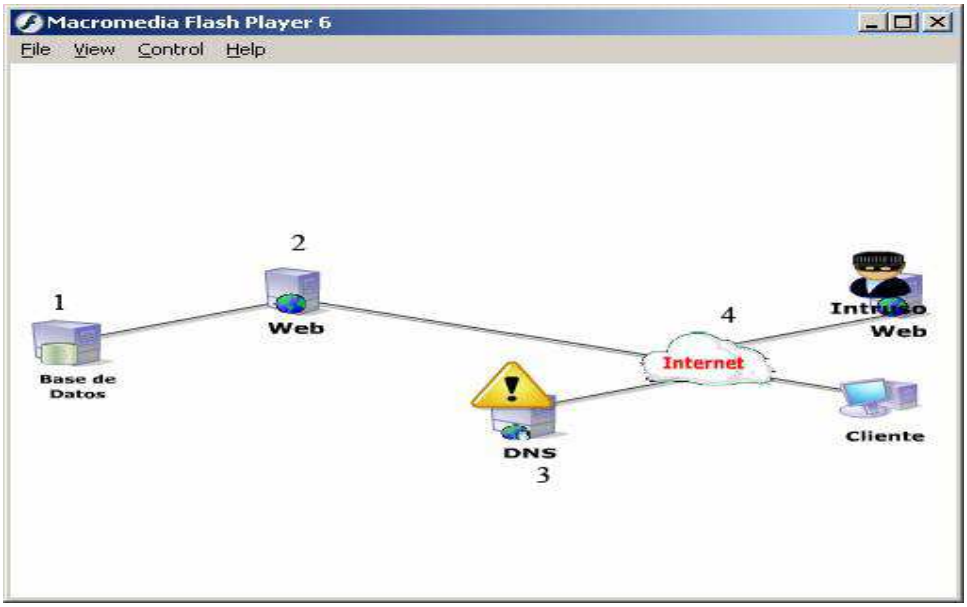


FIG.4.1: Una red en amenaza.

4.1.5 Firewall

El funcionamiento de éste tipo de programas se basa en el "filtrado de paquetes". Todo dato o información que circule entre una PC y la Red es analizado por el programa

(firewall) con la misión de permitir o denegar su paso en ambas direcciones (Internet-->PC ó PC---Internet), como lo muestra la figura 4.2, en la que el intruso intenta acceder a la red y es rechazado por la pared de fuego.

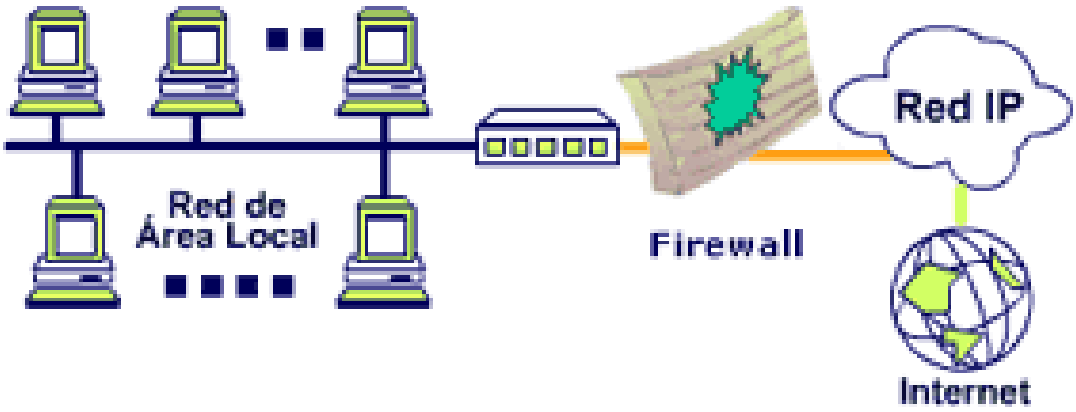


FIG 4.2: Diagrama de un Firewall

4.2 Recursos

Para el desarrollo del proyecto se requiere una PC conectada a Internet, artículos, libros, revistas con respecto a la seguridad en aplicaciones Web, también se requerirá los siguientes productos:

- Lenguaje Java
- MySQL
- Tomcat
- Firewall
- Verisign

4.2.1 Tomcat

Tomcat es un servidor Web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor Web Apache.

Tomcat puede funcionar como servidor Web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Tomcat es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software Licence*. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 6.x, que implementan las especificaciones de Servlet 2.4 y de JSP 2.0. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

4.2.1.1 Estructura de Directorios

La jerarquía de directorios de instalación de Tomcat incluye:

- bin - arranque, cierre, y otros scripts y ejecutables
- common - clases comunes que pueden utilizar Catalina y las aplicaciones Web
- conf - ficheros XML y los correspondientes DTD para la configuración de Tomcat

- logs - logs de Catalina y de las aplicaciones
- server - clases utilizadas solamente por Catalina
- shared - clases compartidas por todas las aplicaciones web
- webapps - directorio que contiene las aplicaciones web
- work - almacenamiento temporal de ficheros y directorios

Tomcat empezó siendo una implementación de la especificación de los servlets comenzada por James Duncan Davidson, que trabajaba como arquitecto de software en Sun y que posteriormente ayudó a hacer el proyecto *open source* y en su donación a la Apache Software Foundation.

4.2.2 Firewall

4.2.2.1 Cortafuegos de capa de red o de filtrado de paquetes

Funciona a nivel de red (nivel 3) de la pila de protocolos (TCP/IP) como filtro de paquetes IP. A este nivel se pueden realizar filtros según los distintos campos de los paquetes IP: dirección IP origen, dirección IP destino. A menudo en este tipo de cortafuegos se permiten filtrados según campos de nivel de transporte (nivel 4) como el puerto origen y destino, o a nivel de enlace de datos (nivel 2) como la dirección MAC.

4.2.2.2 Cortafuegos de capa de aplicación

Trabaja en el nivel de aplicación (nivel 7) de manera que los filtrados se pueden adaptar a características propias de los protocolos de este nivel. Por ejemplo, si se trata de tráfico HTTP se pueden realizar filtrados según la URL a la que se está intentando acceder. Un cortafuegos a nivel 7 de tráfico HTTP es normalmente denominado Proxy y permite que los computadores de una organización entren a Internet de una forma controlada.

4.2.2.3 Cortafuegos personal

Es un caso particular de cortafuegos que se instala como software en un computador, filtrando las comunicaciones entre dicho computador y el resto de la red y viceversa

4.2.2.4 Limitaciones de un cortafuego.

Un cortafuegos no puede protegerse contra aquellos ataques que se efectúen fuera de su punto de operación.

El cortafuegos no puede protegerse de las amenazas a que esta sometido por traidores o usuarios inconscientes. El cortafuegos no puede prohibir que los traidores o espías corporativos copien datos sensibles en disquetes o tarjetas PCMCIA y sustraigan éstas del edificio.

El cortafuegos no puede proteger contra los ataques de la “Ingeniería social”

El cortafuegos no puede protegerse contra los ataques posibles a la red interna por virus informáticos a través de archivos y software. La solución real esta en que la organización debe ser consciente en instalar software antivirus en cada máquina para protegerse de los virus que llegan por medio de disquetes o cualquier otra fuente.

El cortafuego no protege de los fallos de seguridad de los servicios y protocolos de los cuales se permita el tráfico. Hay que configurar correctamente y cuidar la seguridad de los servicios que se publiquen a Internet

4.2.2.5 Políticas de un cortafuego

Hay dos políticas básicas en la configuración de unos cortafuegos y que cambian radicalmente la filosofía fundamental de la seguridad en la organización:

Política restrictiva: Se deniega todo el tráfico excepto el que está explícitamente permitido. El cortafuego obstruye todo el tráfico y hay que habilitar expresamente el tráfico de los servicios que se necesiten.

Política permisiva: Se permite todo el tráfico excepto el que esté explícitamente denegado. Cada servicio potencialmente peligroso necesitará ser aislado básicamente caso por caso, mientras que el resto del tráfico no será filtrado.

La política restrictiva es la más segura, ya que es más difícil permitir por error tráfico potencialmente peligroso, mientras que en la política permisiva es posible que no se haya contemplado algún caso de tráfico peligroso y sea permitido por defecto Un firewall es un dispositivo que funciona como cortafuegos entre redes, permitiendo o denegando las transmisiones de una red a la otra. Un uso típico es situarlo entre una red local y la red Internet, como dispositivo de seguridad para evitar que los intrusos puedan acceder a información confidencial.

Un firewall es simplemente un filtro que controla todas las comunicaciones que pasan de una red a la otra y en función de lo que sean permite o deniega su paso. Para permitir o denegar una comunicación el firewall examina el tipo de servicio al que corresponde, como pueden ser el web, el correo o el IRC. Dependiendo del servicio el firewall decide si lo permite o no. Además, el firewall examina si la comunicación es entrante o saliente y dependiendo de su dirección puede permitirla o no.

De este modo un firewall puede permitir desde una red local hacia Internet servicios de web, correo y ftp, pero no a IRC que puede ser innecesario para nuestro trabajo. También podemos configurar los accesos que se hagan desde Internet hacia la red local y podemos denegarlos todos o permitir algunos servicios como el de la web, (si es que poseemos un servidor web y queremos que accesible desde Internet). Dependiendo del firewall que tengamos también podremos permitir algunos accesos a la red local desde Internet si el usuario se ha autentificado como usuario de la red local.

Un firewall puede ser un dispositivo software o hardware, es decir, un aparatito que se conecta entre la red y el cable de la conexión a Internet, o bien un programa que se instala en la máquina que tiene el modem que conecta con Internet. Incluso podemos encontrar ordenadores computadores muy potentes y con softwares específicos que lo único que hacen es monitorizar las comunicaciones entre redes.

4.2.3 B.D. MySQL [9]

SQL (*Lenguaje de Consulta Estructurado*) fue comercializado por primera vez en 1981 por IBM, el cual fue presentado a ANSI y desde ese entonces ha sido considerado como un estándar para las bases de datos relacionales. Desde 1986, el estándar SQL ha aparecido en diferentes versiones como por ejemplo: SQL:92, SQL:99, SQL:2003. MySQL es una idea originaria de la empresa opensource MySQL AB establecida inicialmente en Suecia en 1995 y cuyos fundadores son David Axmark, Allan Larsson, y Michael "Monty" Widenius. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad.

Michael Widenius en la década de los 90 trató de usar mSQL para conectar las tablas usando rutinas de bajo nivel ISAM, sin embargo, mSQL no era rápido y flexible para sus necesidades. Esto lo conllevó a crear una API SQL denominada MySQL para bases de datos muy similar a la de mSQL pero más portable.

La procedencia del nombre de MySQL no es clara. Por más de 10 años, las herramientas han mantenido el prefijo My. También, se cree que tiene relación con el nombre de la hija del cofundador Monty Widenius quien se llama My.

Por otro lado, el nombre del delfín de MySQL es Sakila y fue seleccionado por los fundadores de MySQL AB en el concurso “Name the Dolphin”. Este nombre fue enviado por Ambrose Twebaze, un desarrollador de Opensource Africano, derivado del idioma SiSwate, el idioma local de Swaziland y corresponde al nombre de una ciudad en Arusha, Tanzania, cerca de Uganda la ciudad origen de Ambrose.

4.2.3.1 Lenguaje de programación

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, REALbasic (Mac), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que

permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

4.2.3.2 Aplicaciones

MySQL es muy utilizado en aplicaciones web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

4.2.3.3 Especificaciones y plataforma.

MySQL funciona sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows. También existe MySQL para OpenVMS en http://www.pi-net.dyndns.org/anonymous/kits/.

4.2.3.4 Características de la versión 5.0.++

Un amplio subconjunto de ANSI SQL 99, y varias extensiones.
Soporte a multiplataforma
Procedimientos almacenados
Triggers
Cursors
Vistas actualizables
Soporte a VARCHAR
INFORMATION_SCHEMA

Modo Strict

Soporte X/Open XA de transacciones distribuidas; transacción en dos fases como parte de esto, utilizando el motor InnoDB de Oracle y motores de almacenamiento independientes (MyISAM para lecturas rápidas, InnoDB para transacciones e integridad referencial).

Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación(savepoints) con InnoDB

Soporte para SSL
Query caching
Sub-SELECTs (o SELECTs anidados)

Buscando campos de texto completos usando el motor de almacenamiento MyISAM

4.2.3.5 Características adicionales

GNU Automake, Autoconf, y Libtool para portabilidad

Uso de multihilos mediante hilos del kernel.

Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice

Tablas hash en memoria temporales

El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL

Completo soporte para operadores y funciones en cláusulas select y where.

Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación.

Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está encriptado al conectarse a un servidor.

Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.

Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).

Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.

En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.

MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.

Disponibilidad en gran cantidad de plataformas y sistemas.

Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.

Transacciones y claves foráneas.

Conectividad segura.

Replicación.

Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GLP o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

Mejoras futuras

El mapa de ruta de MySQL 5.1 indica soporte para:

Particionado de la base de datos

Backup en línea para todos los motores de almacenamiento

Replicación segura

Restricciones a nivel de columna

Planificación de eventos

Funciones XML

4.2.3.6 Características distintivas

Las siguientes características son implementadas únicamente por MySQL:

Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.

Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Tipos de compilación del servidor

Hay tres tipos de compilación del servidor MySQL:

Estándar: Los binarios estándar de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.

Max (No se trata de MaxDB, que es una cooperación con SAP): Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.

MySQL-Debug: Son binarios que han sido compilados con información de depuración extra. No debe ser usada en sistemas en producción porqué el código de depuración puede reducir el rendimiento.

Especificaciones del código fuente

MySQL está escrito en una mezcla de C y C++. Hay un documento que describe algunas de sus estructuras internas en http://dev.mysql.com/doc/internals/en/ (en inglés).

4.2.4 Verisign

Los certificados digitales, tienen una similitud con las licencias de conducir, las primeras permiten viajar por las carreteras, los certificados digitales permiten navegar por Internet, la principal característica es que da identidad al usuario y puede navegar con seguridad. De igual forma que la licencia de conducir o un pasaporte sirve para dar identidad a quien la porta en ciertos casos, el certificado digital da identidad a una clave pública y se comporta como una persona en el espacio cibernético.
El nacimiento del certificado digital fue a raíz de resolver el problema de administrar las claves públicas y que la identidad del dueño no pueda ser falsificada. La idea es que una tercera entidad intervenga en la administración de las claves públicas y asegure que las claves públicas tengan asociado un usuario claramente identificado. Esto fue inicialmente planteado por Kohnfelder del MIT en su tesis de licenciatura.

4.2.4.1 Partes importantes de certificado digital.

Las tres partes más importantes de un certificado digital son:

- Una clave pública
- La identidad del implicado: nombre y datos generales,
- La firma privada de una tercera entidad llamada autoridad certificadora que todos reconocen como tal y que válida la asociación de la clave pública en cuestión con el tipo que dice ser.

En la actualidad casi todas las aplicaciones de comercio electrónico y transacciones seguras requieren un certificado digital, se ha propagado tanto su uso que se tiene ya un formato estándar de certificado digital, este es conocido como X509 v. 3

Algunos de los datos más importantes de este formato son los siguientes:

Versión: 1,2 o 3
Número de Serie: 0000000000000000
Emisor del Certificado: VeriSign
Identificador del Algoritmo usado en la firma: RSA, DSA o CE
Periodo de Validez: De Enero 2009 a Dic 2009
Sujeto: Antonio Jerónimo Ramos
Información de la clave pública del sujeto: la clave, longitud, y demás parámetros
Algunos datos opcionales, extensiones que permite la v3
Firma de la Autoridad Certificadora

Un certificado digital entonces se reduce a un archivo de uno o dos koctetos de tamaño, que autentifica a un usuario de la red.

En una aplicación un certificado digital se puede ver como en la siguiente figura 4.3

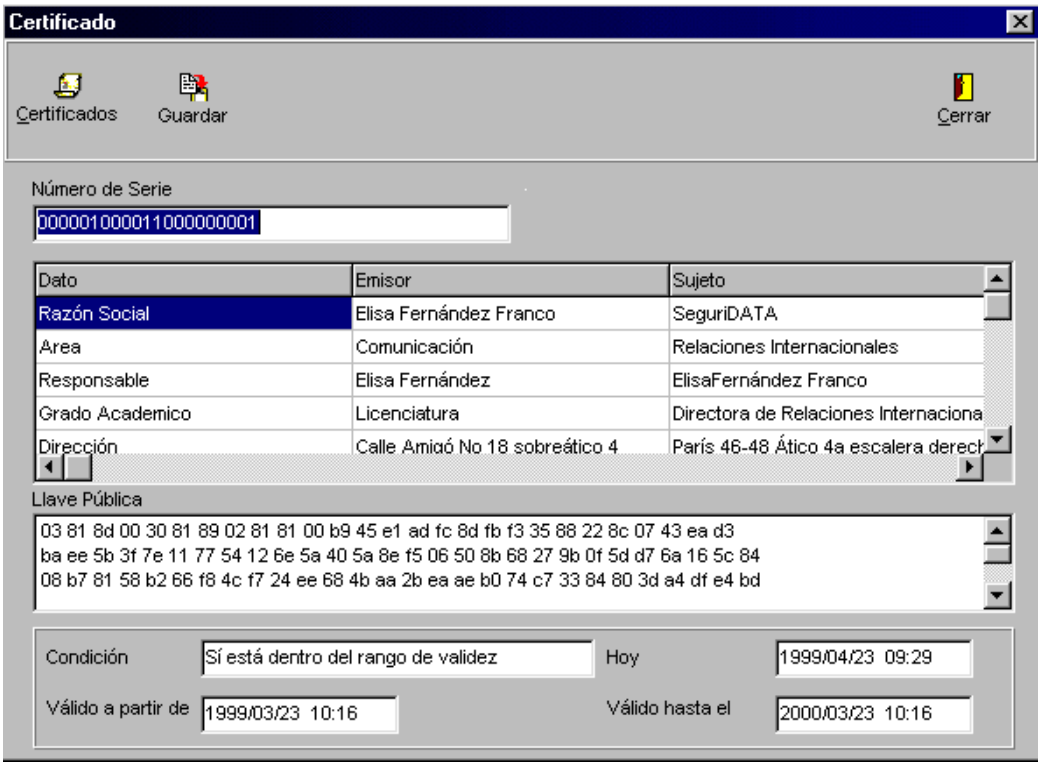


FIG.4.3: Certificado digital.

CAPÍTULO 5

CONCLUSIONES

El capitulo presenta las conclusiones.

5.1 Conclusiones

La seguridad en el manejo de la información es hoy tan importante que la propuesta que en este trabajo se genera, intenta ser un apoyo en cualquiera de los niveles antes mencionados, pero sobre todo a nivel de empresas que es donde se pueden presentar problemas de valor incalculable en lo económico sin dejar de lado la importancia del tiempo y desinformación que pueden generar agentes externos.

Se requieren aplicaciones seguros donde se comparte información confidencial a nivel empresarial, y donde se realicen negocios por medio del Internet como es el comercio electrónico y el negocio entre empresas.

La seguridad en aplicaciones web no puede limitarse a un punto de entrada o a un único elemento de la red. La clave de una seguridad efectiva pasa por ofrecer una cobertura total sobre los puntos de la red, las aplicaciones y todo el tráfico que discurre por ella. Como ya hemos señalado en esta propuesta, la integración de las diferentes comunicaciones de las empresas y de sus redes permite que se reduzcan los costos de mantenimiento, y la integración de la seguridad de todas las comunicaciones es una prueba de ello.

La solución mas adecuada pasa por una seguridad integrada que cubra todos los extremos de la comunicación, lo que tradicionalmente se conoce como una solución *end to end*, que capacite a las organizaciones, además, para medida que vayan apareciendo en le mercado. No se trata de instalar elementos de seguridad y olvidarse, sino de tener una política de mantenimiento y actualización que nos permita trabajar sobre un entorno seguro.

Bibliografía

PAGINA

[1] http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/SEGSO200_archivos/gusano.htm

[2] http://www.albanet.com.mx/articulos/HISTORIA.htm
http://www.svn.net/datamonk/sjmn.05.04.95.html
http://www.porcupine.org/auditing/

[3] http://www.sindominio.net/biblioweb/telematica/hacker-como.html
http://www.perantivirus.com/sosvirus/hackers/index.htm

[4] http://www.prnewswire.co.uk/cgi/news/release?id=131151
http://www.feste.org/
http://www.htmlweb.net/seguridad/varios/firma_certificados.html
http://www.zonagratuita.com/servicios/seguridad/firmae.html

[5] http://es.wikipedia.org/wiki/Firewall
http://www.howstuffworks.com/firewall.htm
http://www.firewalls.com/

[6] http://tomcatbook.sourceforge.net/es/proj_intro.shtml
http://tomcat.apache.org/

[7] http://www.htmlweb.net/seguridad/ssl/ssl_3.html
http://www.verisign.com/

[8] http://www.programacion.com/java/
http://java.sun.com/

[9] http://es.tldp.org/Manuales-LuCAS/manual_PHP/manual_PHP/
http://www.mysql-hispano.org/
http://es.tldp.org/Manuales-LuCAS/manual_PHP/manual_PHP/

[10] http://recursostic.javeriana.edu.co/wiki/index.php/Historia_de_las_bases
www.codigolibre.org/modules.php?name=News&file=print&sid=3299

[11] Sams Teach Yourself MySQL® in 10 Minutes

[12] MySQL in a Nutshell

[13] MySQL AB's JDBC Driver for MySQL

[14] http://www.monografias.com/trabajos7/kerbe/kerbe.shtml

[15] http://www.laflecha.net/canales/seguridad/200710011
http://www.ciudadfutura.com/mundopc/cursos/firewalls/fire1.htm

[16] http://web.archive.org/web/20050217214601/http://www.libelum.com/default.asp?Id=34&Fd=2

[17] http://www.conocimientosweb.net/dcmt/ficha12298.html

[18] http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/node274.html

[19] http://www.configurarequipos.com/truco367.html

[20] http://technet2.microsoft.com/WindowsServer/es/Library/9f9a2ac0-55ef-41bf-8e9e-59effe6af7893082.mspx?mfr=true

[21] http://www.mysqldevelopment.com MySQL development

[22] http://www.planetmysql.org MySQL Weblogs

[23] http://www.db4free.net Free MySQL 5 provider

[24] http://db4free.blogspot.com/ Markus Popp's blog

[25] http://rpbouman.blogspot.com/ Roland Bouman's blog

[26] http://www.futhark.ch/mysql/ Beat Vontobel's blog

[27] http://datacharmer.blogspot.com/ Giuseppe Maxia's blog

[28] http://www.jpipes.com/ Jay Pipe's blog

[29] http://mike.kruckenberg.com/ Mike Kruckenberg's blog

[30] http://sheeri.com/ Sheeri Kritzer's blog

[31] http://mysqldatabaseadministration.blogspot.com/ MySQL

GLOSARIO DE TERMINOS

TCP/IP:	Transmission Control Protocol (TCP) y el Internet Protocol (IP). Protocolo de control de transmisión / Protocolo de Internet.
PSTN:	Public Switched Telephone Network. Red telefónica pública conmutada.
DDoS/Dos:	Distributed Denial Of Service/ Denial Of Service Denegación de servicio distribuido / denegación de servicio
WWW:	Internet.
URL:	Uniform Resource Locutor localizador uniforme de recurso
FTP:	File Transfer Protocol Protocolo de transferencia de archivo
CERT	Equipo de Respuesta a Incidentes de Seguridad en Cómputo
IAB	Internet Architecture Board,
PC	personal computer, computadora personal
BD	Base de datos
DNS	Domain name server, Servidor de nombres de dominio
SABD	Sistema Administrador de Base de Datos
FESTE	Fundación para el Estudio de la Seguridad en las Telecomunicaciones.
IPSCA	IPS Certification Authority,
ACE	Agencia de Certificación electrónica
UDP	User Datagram Protocol.
MITM	Man-in-the-middle
TELNET	Telecommunication Network,
SSL	Secure Socket Layer

SSH	Secure Shell
OS	Operative system, sistema operativo
IBM	internacional bussines machine, negocio internacional de maquinas
HTTP	Hypertext transfer protocol, protocolo de transferencia de hipertexto
HTML	Hypertext markup language, Lenguaje de marcado de hipertexto.
PHP	Hypertext Pre-processor, preprocesador de hipertexto.
ASP	Active Server Pages , servidor de paginas activas
CGI	Common Gateway Interface
JSP	Java Server pages, servidor de paginas java
TLD	Top Level Domain, dominio de alto nivel
ORG	Organismo
COM	Comercial
EDU	Educacion
NET	Red
NAT	Network Address Translation, traductor de direcciones red
PCMCIA	Personal Computer Memory Card International Association, Asociación internacional deTarjetas de memoria para computadoras personales