CAPÍTULO 1.

AMENAZAS A LA SEGURIDAD EN REDES

Afortunadamente, los actos de vandalismo computacional significan una mucho menor amenaza a los planes de seguridad computacional, que una pobre política de respaldos o un deficiente plan de contingencia. Pero una persona activamente esforzándose en un acto de vandalismo computacional puede causar enormes desastres. El propósito de este capítulo es llevar a cabo una revisión minuciosa de los métodos empleados por estos vándalos para penetrar e interrumpir los sistemas de cómputo. No se pretende legitimar a ninguno de ellos, tan sólo lograr una diferenciación objetiva y útil para su identificación y poder protegerse de ellos.

La seguridad de un sistema depende de la gente que tiene la posibilidad de accesarlo. Puede estar totalmente desprotegido, de forma tal que la buena operación continua sea importante para la gente que lo accesa, asumiendo que todos ellos son responsables, y los respaldos regulares son realizados sólo para casos de problemas de hardware.

En la actualidad, debido a la gran afluencia de las redes a Internet se pueden presentar diversos problemas como son:

- El Internet expone las computadoras conectadas a muchos programas destructivos.
- Las pérdidas de corporaciones debido a ataques computacionales se han incrementado.
- Existen muchos riesgos en cada red de computadoras, pero es porque están expuestas a un numero mucho mayor de hackers potenciales, es decir, las computadoras conectadas a internet tienen más riesgos que las que están conectadas internamente o privadamente.

1.1 Incremento del riesgo.

El vandalismo computacional se manifiesta en muchas formas. Las diferentes categorías de este vandalismo se basan en cómo éstas se extienden y se activan. Los programas **Caballos de Troya**, por ejemplo, son programas disfrazados como algo inofensivo pero activados por la propia víctima. [HackPr97]

Los programas de **virus** modifican otros programas (siempre causando resultados desastrosos), mientas se duplican a sí mismos y buscan formas para extenderse a otros sistemas. [HackPr97]

Los programas **bacteria** crean copias de sí mismos en forma geométrica, siendo su modo primario de dañar por medio de consumir recursos computacionales hasta que el sistema llega a una paralización. [HackPr97]

Los **gusanos** son programas únicos que migran de computadora a computadora sobre la red, mientas van dañando al sistema o divulgando información crítica del sistema a sus creadores, con el fin de preparar el camino a ataques más directos. [HackPr97]

Las **puertas traseras** son aspectos no documentados construidos dentro de programas y que pueden proveer a usuarios con conocimientos, un acceso no autorizado a los recursos computacionales. [HackPr97]

Las **bombas lógicas** son programas diseñados para dañar un sistema y se activan por cambios futuros en la configuración del mismo. [HackPr97]

Las **trampas** son también aspectos no documentados construidos dentro de los programas, activados por usuarios involuntarios, que trastornan la computadora. Colectivamente estas amenazas son conocidas como **amenazas programadas**. [HackPr97]

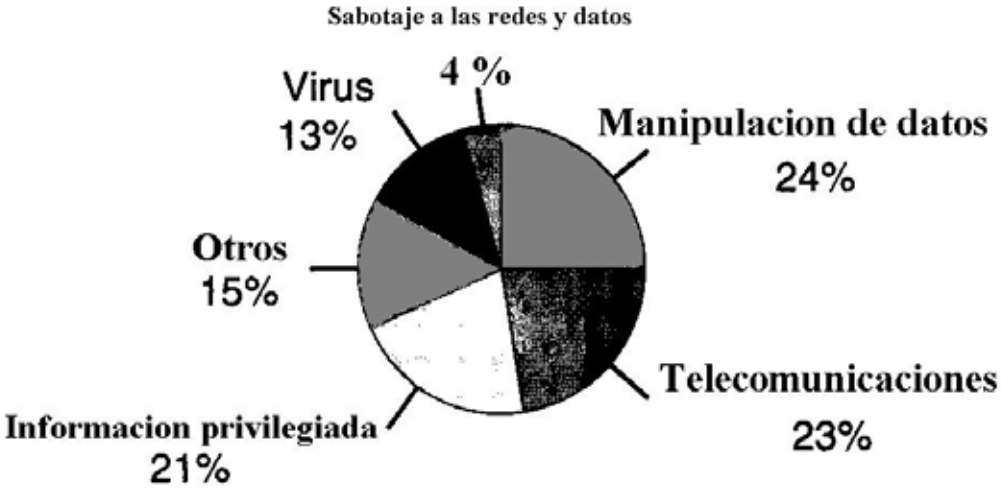

Figura 1.1 Muestra una gráfica del total de pérdidas de acuerdo con la naturaleza de los ataques. [HackPr97]

1.2 Considerando los riesgos

En Internet existen muchas personas que se dedican a robar y destruir todo tipo de información. Estos pueden ser gente de la misma compañía o empresa que tienen fácil acceso a la red y también existen personas externas que logran violar la seguridad de la red. Estas personas son muy peligrosas ya que pueden robar cualquier cosa que este en las computadoras y los usuarios o administradores pueden darse cuenta del ataque después de varias semanas e incluso meses.

En ocasiones una empresa no se da cuenta del robo de su información hasta que se presentan las pérdidas de clientes e ingresos por que la competencia tiene información vital para llevar la delantera.

En el mundo real se tienen ciertas medidas de seguridad como es el cerrar con llave el automóvil cuando lo estacionamos, en el mundo virtual se debe asegurar la red cuando nos unimos a Internet. La red mundial está plagada de personas que, ya sea por dinero o por puro gusto, destruirá la seguridad de los equipos, bajara información confidencial y robará o destruirá la información.

A estas personas se les conoce comúnmente como *hackers*. Estos son personas que se dedican a irrumpir los equipos por placer o por algún beneficio económico por parte quien los contrata.

Dentro de los peligros que podemos encontrar tenemos los siguientes que son considerados los más importantes ya que son los más usados y conocidos:

1.3 Virus

Un virus es un programa destructivo que modifica otros programas insertando copias de sí mismo, en un esfuerzo por ocultar su existencia y propagarse a sí mismo en la red. Esta es una forma molesta de ataque en el sistema por que se comporta como un parásito. Cuando el programa infectado es ejecutado, también se ejecuta el código viral. Aunque, dependiendo de la naturaleza de los virus, el código original puede o no puede ser ejecutado. [TomVir]

Los virus no pueden ejecutarse como un programa independiente; ellos necesitan un programa anfitrión (host program) que los inicialice. Una vez que el virus se ha establecido y atacado a otros programas en el sistema, es difícil eliminarlo.

Un virus computacional comparte muchos de los atributos de los biológicos convencionales. Consiste de tres subsistemas: mecanismo de infección, activador (trigger) y misión. [TomVir]

No todos los virus son perjudiciales para un sistema, un virus benéfico por ejemplo, podría comprimir todos los programas en un sistema para conservar espacio en disco y descomprimirlos cuando son ejecutados, permitiendo al programa, llegar a ser ejecutable otra vez. [HackPr97]

Las siguientes recomendaciones ayudan a proteger los sistemas de virus computacionales:

- Centralizar la responsabilidad de mover cualquier archivo entre sistemas, con el fin de proveer un estricto control e inspección minuciosa.

- Implantar una política de respaldos con respaldos completos del sistema almacenados por largo tiempo (así que el software más estable del sistema operativo y el software de terceros pueda ser recuperado de medios sin infección como parte de la reconstrucción del sistema).

- Mantener los archivos temporales fuera de los directorios del sistema operativo y de los que soportan productos de software de terceros. Esto no necesariamente protege al sistema y archivos de terceros de una infección externa, sin embargo hace más ordenada la recuperación, así como reduce la oportunidad de que el virus sea reintroducido en el sistema.

- Mantener actualizado con la literatura de UNIX para estar atento de epidemias de virus en la comunidad de UNIX.

- Establecer máscaras de usuarios (umasks) para que los programas escritos por usuarios no puedan ser invadidos por virus que tengan permisos insuficientes.

- Proteger directorios de forma tal que estos no sean fácilmente contaminados por virus.

- Desarrollar políticas para el uso de grupos de usuarios en sistemas UNIX, así que un virus proveniente de group ID público no sea capaz de infectar programas compartidos con otros grupos de usuarios.

1.4 Gusanos (Worms)

Los worms (gusanos) son programas autoreplicables y autoinicializables, diseminables por ellos mismos de máquina en máquina a través de arrastrarse por la red. Aprovechan los “security holes” (huecos de seguridad) conocidos. Un worm no altera o daña otros programas, pero podría ser un vehículo para otros programas como los virus.

Un worm no necesariamente verifica la máquina atacada para ver si ya está contaminada. Puede causar un rechazo de los servicios por estar usando todo el espacio en disco. Algunas veces estos programas son diseñados para enviar simplemente de regreso al desarrollador información acerca de los sistemas, la cual puede ser usada más tarde para atacar al sistema directamente, y otras veces ellos pueden hacer daño en su trayecto (posiblemente dejando una bacteria o virus en su camino).

Generalmente estas entidades de red gastan mucho de su tiempo recogiendo y procesando archivos de seguridad y de red, intentando encontrar rutas en la misma hacia otros sistemas e intentando adivinar passwords.

Un worm consiste de tres partes: búsqueda de un nuevo host para infectarlo, copia de sí mismo al nuevo host y provocar que la nueva copia sea ejecutada.

Los síntomas del ataque de un gusano se pueden apreciar en los archivos de log (tales como su.log, el cual indicará los numerosos intentos sin éxito de una entidad no autorizada para convertirse en superusuario), significante incremento en el tráfico de la red (lo cual se manifiesta como una reducción en la capacidad de procesamiento normal), y procesos anormales corriendo en el sistema (los cuales pueden ser desplegados mediante le comando ps – process status). [HackPr97]

1.5 Puertas Traseras (Back Doors)

Las back doors (puertas traseras) son conocidas también como trap doors (trampas), aunque entre ellos existen diferencias importantes que se describen en este capitulo. Son programas o partes de programa que permiten el acceso no autorizado a un sistema. Algunas veces son insertados maliciosamente en los sistemas, aunque otras los programadores y desarrolladores los escriben usualmente en aplicaciones que requerirán amplios procedimientos de autentificación.

Los back doors permiten al usuario entrar a los programas rápidamente para propósitos de evaluación, depuración, mantenimiento y monitoreo en el proceso de desarrollo de sistemas. Muchas veces los back doors son olvidados y dejados en el código cuando éste es liberado. Potencialmente destructivos los back doors pueden existir en programas por muchos años antes de ser descubiertos.

Los back doors pueden presentar problemas cuando son descubiertos por hackers sin escrúpulos. Es por eso que se consideran una amenaza real a la seguridad del sistema. Uno de los aspectos más significativos de esta amenaza es que se encuentran disponibles para muchos usuarios. Más que requerir un grado particular de conocimientos técnicos y destreza, para estas amenazas se necesita conocer el back door y puede ser fácilmente pasado de boca en boca o enviado por correo electrónico en bulletin boards.

La mejor defensa contra un ataque a través de una puerta trasera es obteniendo el conocimiento de ésta, antes de que llegue a ser ampliamente difundida. Por lo cual, una de las mejores protecciones es la comunicación entre administradores de sistemas. [HackPr97]

1.6 Trampas (Trap Doors)

Las trampas son actualmente consideradas como un caso especial de bomba lógica, aunque se parecen a las puertas traseras, dado que son aspectos no documentados o modos de operación de programas que de otra forma son confiables. Sin embargo, mientras que las back doors son deliberadamente explotadas por usuarios conocedores, las trap doors son disparadas por algún conjunto de condiciones de habilitación causando que estas realicen sus acciones destructivas. Estas condiciones podrían ser la hora del sistema o la identificación del usuario al momento de ejecutar un programa. [HackPr97]

1.7 Bombas Lógicas (Logic bombs)

Las bombas lógicas son características ocultas construidas en un programa ejecutadas cuando se cumplen ciertas condiciones, tales como, un cierto conjunto de claves, o cierta fecha alcanzada, modificando dramáticamente su comportamiento.

Las bombas lógicas ejecutan una función, o un conjunto de funciones, que no fueron características intencionales del programa original, siendo las más comunes la destrucción de aplicaciones o datos. Son frecuentemente colocadas por programadores encargados de mantenimiento de sistemas. El famoso virus Miguel Angel, fue disparado por una bomba lógica.

Existen muchos usos legítimos de bombas lógicas. Los time-out son ampliamente usados por los vendedores de software, permiten administrar las provisiones contractuales o reforzar agendas de pago. La ejecución de una bomba lógica no necesariamente es disparada por el reloj.

Las bombas lógicas son frecuentemente perpetradas no por personas ajenas al sistema quienes han ganado acceso no autorizado (ya que ellos prefieren hacer el daño tan pronto como sea posible), sino por usuarios quienes están autorizados para tener acceso al sistema.

Un caso documentado es la bomba lógica de Michael J. Lauffenburger insertada en un programa llamado Cleanup el 20 de Marzo de 1991, la cual esta dispuesta para activarse le 24 de Mayo a las 6:00 PM., siendo sus funciones eliminar el programa de seguimiento (PTP), borrar la base de datos (SAS.DB) y autodestruírse sin dejar una huella. Un compañero de trabajo la descubrió accidentalmente el 10 de Abril. Finalmente, Michael fue arrestado el 31 de Abril.

La mejor protección contra los desastres de las bombas lógicas es tener bien definidos procesos de administración y mantenimiento de cuentas de usuario. Tales procedimientos serán enfocados para detectar bombas lógicas antes de que éstas tengan la oportunidad de hacer daños. [HackPr97]

1.8 Caballo de Troya (Troyan horse)

Los Caballos de Troya son probablemente las amenazas programadas más comunes y fáciles de implantar, son programas que imitan a un programa que el usuario quiere ejecutar, pero son realmente diferentes.

Aparentan ser inofensivos, pero permiten violar la seguridad de un sistema, pues se pueden ver como una herramienta estándar de UNIX, aunque hayan sido programados para realizar ciertos actos destructivos cuando se ejecutan por un usuario del sistema con privilegios apropiados. [UnxSec94]

Desafortunadamente el usuario no está siempre consciente de que un Caballo de Troya ha sido ejecutado hasta que el daño se ha realizado. Un Caballo de Troya puede ser usado para capturar passwords, cambiar permisos a archivos o crear programas set-UID.

Un ataque de Caballo de Troya engaña al usuario en la ejecución de un programa dañando al sistema por tomar ventaja de los permisos de acceso del usuario. Otro modo común de ataque es a través del uso de shar (compartir) los archivos respaldados en cinta – archive. Estos archivos son grandes shell scripts que al ejecutarse realizan autoextracciones de archivos, los cuales fueron previamente respaldados en cinta como parte del script en si mismo.

Algunas medidas o hábitos para reducir la oportunidad de que el usuario se convierta en víctima de un Caballo de Troya incluyen lo siguiente: [HackPr97]

- Nunca colocar directorios no estándares (incluyendo **.**o “ “) en un PATH. Probablemente colocarlo en último lugar del PATH el **..**

- Nunca ejecutar el shar de archivos respaldados en cinta, particularmente de procedencia no familiar. Si se necesita ejecutarlos, llevarlo a cabo sólo en un sistema UNIX que no afecte su daño o pérdida total, sino ejecutarlos después de un llamado al sistema chroot que limite el impacto que pueda generarse en el sistema de archivos.

- Proteger todos los editores de archivos de inicialización en el directorio Home, de forma tal que sólo el administrador pueda escribir en ellos. También poner el sticky bit en el directorio Home de forma tal que otros usuarios no tengan permitido borrar archivos que ellos no puedan escribir.

- Si las terminales tienen la habilidad de repetir cadenas de caracteres enviándolos como si fueran escritos en el teclado, deshabilitar este aspecto (o mejor aún , reemplazar la terminal con una que no tenga esta característica).

1.9 Bacterias

Algunas veces también llamadas conejas, son programas que existen para recuperarse a si mismas, y generalmente afectan un sistema por tomar ventaja de los recursos computacionales que ellas consumen sólo por existir en el sistema. Más que pegarse a otros programas, como los virus, las bacterias computacionales simplemente al ser ejecutadas se duplican a si mismas.

La bacteria no altera los datos ni destruye archivos. Su propósito es degradar el servicio del sistema, pues dependiendo de cómo es programada, puede empezar a ocupar todo el espacio en disco o los ciclos de CPU muy rápidamente, llevando al sistema a detenerse. Un programa que es de un solo byte de longitud podría consumir 4 GB. De espacio después de sólo 32 ciclos de reproducción. Los más grandes, programas de medida más real podrían necesitar menos ciclos para sobrecargar el sistema. [HackPr97]

1.10 Huecos de Seguridad (Security Holes)

Los huecos de seguridad son imperfecciones en el diseño de software, que mal usados, otorgan privilegios a usuarios comunes. La mayoría de los servicios en Internet (FTP, TELNET, SENDMAIL) tienen huecos de seguridad.

Los huecos de seguridad se manifiestan en cuatro formas: [HackPr97]

1. Huecos de Seguridad Físicos. Donde el problema potencial es causado por permitir acceso físico al equipo a personas no autorizadas, donde estas pueden realizar operaciones que no deberían ser capaces de hacer.

2. Huecos de Seguridad de Software. Donde el problema es causado por elementos mal escritos de software privilegiado (daemons, cronjobs) los cuales pueden ser utilizados para realizar cosas que no deberían poder hacer.

3. Huecos de Seguridad por Uso Incompatible. Donde, por falta de experiencia o por errores propios, el Administrador del Sistema ensambla una combinación de hardware y software el cual cuando se usa como un sistema está seriamente dañado, desde el punto de vista de la seguridad. Es precisamente esta incompatibilidad de tratar de hacer que dos cosas no conectables pero útiles se integren lo que crea un hueco de seguridad.

4. Selección de una filosofía de seguridad y su mantenimiento. Este hueco de seguridad se manifiesta como un problema de percepción y entendimiento. El software perfecto, el hardware protegido y los componentes compatibles no trabajarán adecuadamente a menos que se seleccione una política de seguridad apropiada y las partes del sistema se direccionen para reforzarla. Pues aún teniendo el mejor mecanismo de password en el mundo, es tiempo perdido si los usuarios piensan que su nombre al revés es un buen password.

Nuevos huecos como estos son descubiertos y lo mejor que se puede hacer es:

- Tratar de estructurar el sistema de forma tal que el menor número de programas de software posible se ejecute con privilegios de ***root/daemon/bin***, los cuales sean conocidos por su robustez.

- Suscribirse a foros donde se obtengan detalles de problemas y soluciones que se apliquen tan rápido como sea posible.

1.11 Insectos (BUGS)

Un Bug es un defecto en un programa que causa que este realice algo inesperado. Estos bugs a menudo son destructivos. Programas escritos en lenguajes de bajo nivel como C o Lenguaje Ensamblador, son especialmente indefensos para los bugs destructivos porque los errores en el direccionamiento de memoria, pueden resultar en sobreescribir datos almacenados en áreas usualmente reservadas para el sistema operativo.

Sin pensar que lenguajes como C o Ensamblador sean malos, es muy importante que los programadores tomen en cuenta que programas mal escritos, pueden resultar desastrosos. [HackPr97]

1.12 Piratas Informáticos (Hackers)

Cuando se empezó a popularizar este término se le daba un significado diferente al que ahora conocemos. La definición de ese entonces era, " *programadores brillantes y constructivos que iniciaron la revolución computacional* ". Hacker era una persona que se dedicaba a la creación de nuevos sistemas amigables y accesibles para todos. Se les conocía como gente que le encanta resolver problemas y crear soluciones usando las nuevas tecnologías.

En la actualidad la definición a cambiado para una nueva clase de programadores. Estos programadores que, incluso se les conoce con otros nombres como: "*crackers, phreackers y phreakers*", destruyen los equipos en lugar de ofrecer soluciones para los mismos. Estas personas son altamente solicitadas en la industria del espionaje y sabotaje.

También participan en una especie de competencia por ser el mejor, cuantos más equipos sean intervenidos mucho más reconocimiento tienen entre los otros hackers. Algunas formas en que operan son colapsando sistemas, robando contraseñas y código de programas sólo para producir tantos problemas como sea posible. [HackPr97]

1.13 Los tipos de amenazas

Como ya hemos visto, el conectar tu sistema a Internet, lo expone a numerosas amenazas que se incrementan diariamente. Los tipos más generales de amenazas son: [HackPr97]

- Vulnerabilidad de información
- Vulnerabilidad en el software
- Debilidades en el sistema físico
- Transmisión de debilidades

Las formas y estilos comúnmente usados en ataques realizados vía Internet en redes corporativas, están divididos en 9 categorías: [HackPr97]

- Ataques basados en passwords

- En base a escuchar el tráfico de la red
- Ataques que explotan los accesos confiables
- Basándose en las direcciones IP
- Introduciendo información sin darse cuenta
- Predicción de números secuenciales
- Secuestrando sesiones
- Ataques enfocados a explotar las debilidades de la tecnología
- Explotando el sistema de librerías compartidas

1.13.1 Ataques basados en passwords

Estos son, históricamente, uno de los favoritos para los hackers. Inicialmente, los hackers tratan de entrar a un sistema en la red por medio de teclear un nombre de usuario y contraseña. Esta persona tratara de una contraseña a otra hasta que una de ellas funcione. Sin embargo ahora existen programas que hacen una decodificación o adivinan los passwords mediante una combinación de todas las palabras y letras de diccionarios en varios idiomas con signos de puntuación y números. [HackPr97]

1.13.2 Escuchando el tráfico de la red

Es posiblemente uno de los más difíciles tipos para llevar a cabo, pero es un ataque muy serio cuando se logra en una transacción comercial. Para ello se utiliza el llamado *packet sniffer*, el cual se encargará de interceptar los paquetes que viajan a través de la red, estos pueden contener información confidencial como las claves de usuarios, paquetes de transacciones comerciales con el número de una tarjeta de crédito, e_mail, etc.. El procedimiento es obtener el IP que recibirá el paquete y así cuando pase uno dirigido a ese host, entonces lo copiará para enviarlo al sistema del hacker. [HackPr97]

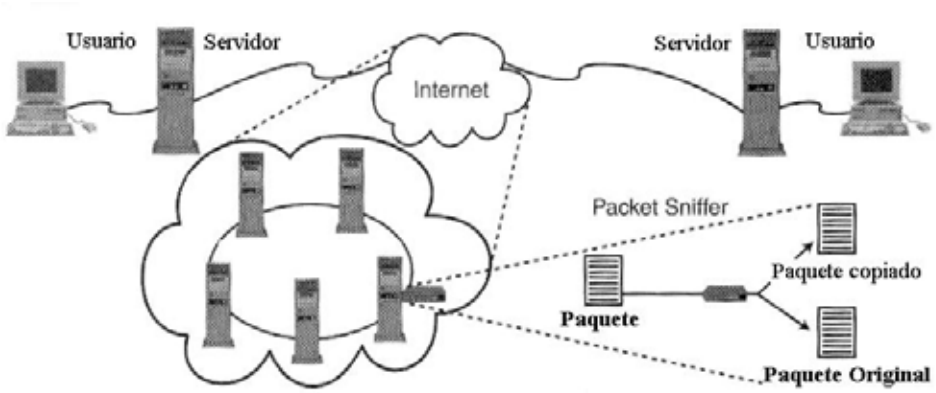

Figura 1.2 Uso del packet sniffer para interceptar los paquetes que viajan por la red. [HackPr97]

1.13.3 Mediante accesos confiables

Son comunes en redes que usan un sistema operativo (incluyendo UNIX, VMS y NT) que incorpora mecanismos de accesos confiables. Los usuarios de estos sistemas pueden crear archivos de hosts confiables (como archivos .rhosts en los directorios base) los cuales incluyen los nombres de máquinas o direcciones IP de las cuales un usuario puede accesar el sistema sin una contraseña para ello. Si un hacker obtiene el nombre de la máquina tendrá privilegios de entrar al sistema y la mayoría de ellos sabe que los administradores de UNIX colocan el archivo en el directorio raíz, con esto se moverían como super-usuarios. [HackPr97]

1.13.4 Con direcciones IP

Como ya sabemos, cuando las computadoras se comunican en la red, lo hacen mediante el direccionamiento de paquetes. Estas direcciones son las llamadas *IP Address* que identifican cada computadora en el mundo. Cuando un hacker hace un ataque de esta manera, da información falsa acerca de la identidad de su computadora, es decir, dice que su computadora es una confiable dentro de una red mediante el duplicado de una dirección TCP/IP. Así el intruso gana los paquete de acceso a un sistema y sus servicios. [HackPr97]

1.13.5 Introduciendo información

Este tipo de ataques se han convertido en comunes y mucho más peligrosos en tanto más usuarios se conectan a la red. Un ejemplo simple es cuando un hacker envía un e_mail a los usuarios informando que el administrador de la red es un intruso y le pide que le envíen su password por este medio y evitar el daño. También se puede hacer usando un applet de Java para avisar que tiene un e_mail nuevo y que necesita poner su clave de acceso para revisarlo, este applet crea una ventana familiar a la vista del usuario para ganar su confianza y es así como se logra obtener su clave. Este tipo de ataque es usual en los usuarios que no conocen mucho acerca de las computadoras y las redes, lo mejor para evitar estos problemas es la educación del usuario. [HackPr97]

1.13.6 Predicción de números secuenciales

Es una técnica común para el robo de IP's dentro de las redes UNIX. El principio de cualquier conexión TCP/IP requiere que las dos máquinas intercambien lo que se llama un "*handshake*," o un paquete de inicio el cual incluye *números secuenciales*. Las computadoras usan estos números como parte de cada transmisión durante la conexión. La creación de estos se realiza basándose en los relojes internos de cada computadora. En muchas versiones de UNIX, los números secuenciales obedecen un patrón que es predecible usando un determinado algoritmo. Después de escuchar estos patrones durante cierto tiempo, hechos por conexiones legítimas, un hacker puede predecir en cierta medida la secuencia de números para lograr un handshake no autorizado. [HackPr97]

1.13.7 Secuestro de sesiones

En este tipo, el intruso encuentra una conexión existente entre dos computadoras, generalmente de un servidor y un cliente. Inmediatamente después penetrando a routers desprotegidos o firewalls inadecuados, obtiene los números de direcciones TCP/IP en un intercambio entre las computadoras.

Después el intruso secuestra la sesión del usuario simulando la dirección del usuario. Al lograr esto, el secuestrador se adueña de la sesión y el host desconecta al usuario legítimo y el intruso obtiene libre acceso a los archivos que el usuario podía llegar. Es muy difícil detectar una sesión secuestrada y lo que se puede hacer para evitar esto es por ejemplo, remover cuentas de acceso innecesarias, conseguir parches de seguridad para proteger los routers y los firewalls, también se puede usar el encriptamiento de paquetes. Es muy importante que se tengan estas medidas por que es virtualmente imposible detectar sesiones secuestradas ya que el secuestrador aparece en el sistema como el usuario secuestrado. [HackPr97]

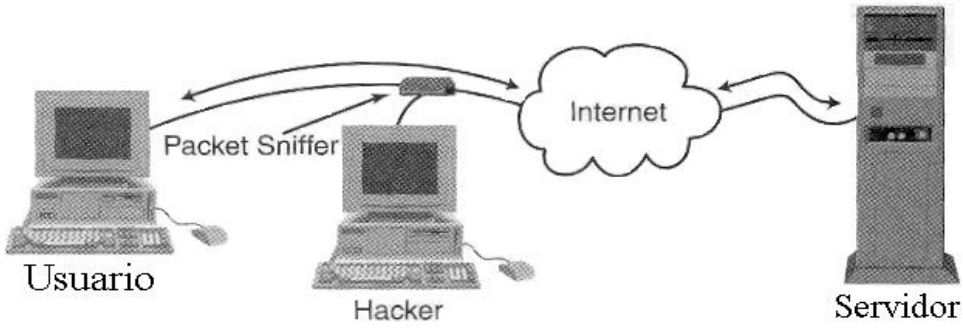

Figura 1.3 El hacker usa el packet sniffer para obtener la dirección IP del usuario o destinatario final [HackPr97]

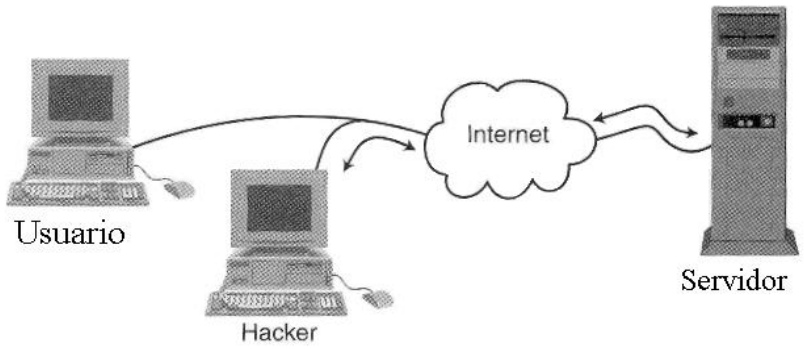

Figura 1.4 El hacker secuestra la sesión fingiendo ser el usuario final y obtiene libre acceso a los archivos del usuario original. [HackPr97].

1.13.8 Explotando las debilidades de la tecnología

Todos los sistemas operativos tienen sus propias debilidades, algunos son más accesibles que otros. Cuando salen los nuevos sistemas pueden contener los llamados bugs que provocarían el colapso de un equipo conectado a la red.

1.13.9 Explotando las librerías compartidas

Esto es muy común en los sistemas UNIX. Una librería compartida es un conjunto de funciones de programas comunes que el sistema operativo carga de un archivo a la memoria RAM en cada petición del programa. [UnxArc93]

Los hackers hacen un reemplazo de estas librerías para sus propósitos, como proveerlos de privilegios para accesar una petición. La solución a este problema es muy simple, se necesita de un buen mantenimiento del sistema de archivos periódicamente y hacer algunas pruebas.

El estudio de estos problemas es importante ya que ofrecen un amplio panorama de lo que los hacker pueden hacer en cualquier intento de ataque a una red. También podemos revisar las características de cada uno de estos tipos te amenazas para prevenir cualquier intrusión al sistema.

Las políticas de seguridad se deben escribir tomando en cuenta todo lo anterior para establecer una línea de fuego. En caso de ser víctimas de un ataque, es necesario tener algún plan de contingencia respecto a la pérdida de información o el implante de algún programa que le permita el acceso al intruso.

Las características de todos los posibles ataques a una red corporativa o institucional nos permiten la creación de una buena política de seguridad. Esta debe contener la mayor cantidad posible de defensas y/o medidas de prevención. Así mismo, este capítulo nos provee de una visión de lo que debemos revisar en cuanto a posibles trampas, o puertas escondidas dentro de los programas que se van a ejecutar dentro de la red.