UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

PROPUESTA PARA IMPLEMENTAR SEGURIDAD EN UNIX, USANDO LA METODOLOGÍA DE ANÁLISIS FORENSE .

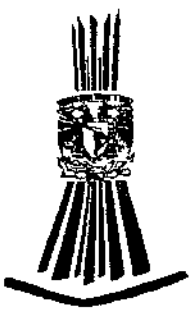
T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

JOSÉ ALFREDO COBIÁN CAMPOS



ASESOR: M. EN C. MARCELO PÉREZ MEDEL

SAN JUAN DE ARAGÓN, EDO. DE MÉXICO 2005

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: José Alfredo Cabiás Campos

FECHA: 21 Junio 05

FIRMA: Ja Cabiás

Agradecimientos

El más grande agradecimiento a la persona que ha luchado con todo su corazón y todo su empeño por hacer de mi lo que soy, mi mamá, Angeles Campos, a quien dedico este trabajo y con el mismo le demuestro que todo su esfuerzo no ha sido en vano.

Deseo dejar constancia de mi más profundo y sincero agradecimiento al M. en C. Marcelo Pérez Medel ya que sin su ayuda, su comprensión y desinteresada colaboración, todo este trabajo habría sido imposible de realizar.

Además resultó invaluable la ayuda brindada por el Dr. Oleg Okunev, M en C Jesus Diaz Barriga Arceo, Silvia Torres Alamilla y el Mat. Salvador Lopez Medoza.

Mi reconocimiento hacia el personal docente de Ingenieria en Computación de la FES Aragón que me incursionaron correctamente, a poder ser un Ingeniero en Computación.

Mi satisfacción y agradecimiento por la comprensión y ayuda recibida por parte de las autoridades de la Universidad Nacional Autonoma de Mexico y a mis amigos que supieron entender la necesidad de la realización de esta tesis.

Sin ánimo de olvidar a nadie en particular y a todos aquellas personas que de una u otra manera han compartido mi vida durante el transcurso de estos últimos años mi más sincero agradecimiento a su comprensión, estímulo y ayuda, ya que todos son parte de mi vida.

Una mención especial a Karla, pues ella me apoyó hasta ver terminado el presente trabajo, formando parte importante en el andar de mi vida.

Índice general

Introducción

En este trabajo se abordará el problema de la comunicación en presencia de atacantes, delincuentes, se estudiarán conceptos básicos de seguridad en cómputo y su entorno. También se hace una reflexión sobre los problemas de la comunidad de multiusuarios de Internet en cuanto a problemas de seguridad. Se hace hincapié en la gran desinformación que existe hoy en día en lo que se refiere a este tema, se establecen soluciones metódicas para su detección y corrección dentro del marco de seguridad y administración mediante políticas adecuadas así como el uso de múltiples herramientas de seguridad. Se analiza un amplio espectro de problemas de seguridad que se presentan en la transmisión y almacenamiento de la información.

Se parte de la premisa de que ningún sistema es computacionalmente seguro, pero se pueden minimizar los brotes de inseguridad administrando y controlando todos los servicios y permisos que son otorgados con el entorno Unix a los usuarios así como a los servicios. En el mercado computacional existe una amplia gama de sistemas operativos, y por considerarlo ventajoso un sector de la población está emigrando a Unix o alguna de sus variantes debido a su trayectoria de más de treinta años de desarrollo. Sin olvidar, que por cuestiones de aplicaciones siempre han convivido más con Unix. Unix cuenta con multiples servicios de red que contemplan la coexistencia con las redes privadas o públicas.

En sectores gubernamentales, de comercio y educación se asignan recursos humanos y técnicos para el fortalecimiento de la seguridad y protección de intereses y servicios dado la comunicación a travdés de Internet, con redes locales en el resto del mundo. No todo los integrantes de la red de Internet conocen el riesgo de ser atacados por gusanos o virus(código computacional que puede van modificar un grupo de programas o documentos), hasta que sufre un incidente de seguridad. Debido a este problema una serie de administradores programadores y desarrolladores tanto de software como de hardware está interesado en desarrollar aplicaciones que no sean portadoras de vulnerabilidades para no arriesgar a sus clientes.

Cada vez es más frecuente encontrar noticias sobre ataques de redes o complejos importantes

que han sido comprometidas por criminales informáticos desconocidos. A pesar de que la prensa argumenta que las intrusiones son obra de adolescentes que querían divertirse, ya no se trata de incidentes aislados o de una institución atacada al azar. A diario se reciben reportes en el CERT[1] como en otros organismos de seguridad, sobre los ataques a redes informáticas.

Los administradores de sistemas computacionales deben utilizar horas y a veces días enteros para volver a instalar o bien reconfigurar sus sistemas comprometidos, con el objeto de recuperar la confianza en la integridad del sistema que administran. No hay forma de saber los motivos que tuvo el intruso para atacar y debe suponerse que sus intenciones siempre pueden ser las peores. Quien irrumpe en los sistemas sin autorización, aunque sea solamente para mirar su estructura, causa mucho daño, incluso sin haber leído correspondencia confidencial y sin borrar ningún archivo. Pero se puede saber que actividades realizó en su estancia.

Los individuos que conviven en dicho entorno hostil aún no se percatan que al hacer una transacción bancaria o comprar en línea pueden ser defraudados o ser victimas en transaciones no deseadas. A los causantes de esas acciones los catalogo como atacantes; dichos atacantes se pueden encuentrar trabajando en la empresa, que van a desfalcar o ser simples visitantes de sitio de internet, pasando por un espectro muy amplio de posibilidades. Sin importar la dimensión o tipo de empresa su activo mas valioso es su información, la pérdida o cualquier daño a su integridad, representa por lo general un riesgo económico para la empresa; por ello, es necesario tener en mente la seguridad de las plataformas, aplicaciones, subredes y usos.

En las noticias es constante encontrar que servidores y redes de computadoras de organizaciones han sido comprometidas por delincuentes informáticos, a pesar de la tecnología y la administración con que cuentan para evitar las intrusiones. Esto, debido a que toda computadora que esté conectada a la red Internet está propensa a ser atacada debido a múltiples factores como la falta de cultura en sistemas de seguridad que tienen los administradores, los usuarios, la vulnerabilidad del software y hardware. No es difícil pensar que puede haber más de una persona con intenciones perversas para penetrar y robar información de una organización sea está publica o privada. Por eso es necesario proteger adecuadamente a las computadoras, a las redes en Internet.

En Internet, los `Hackers` han encontrado "un nuevo gran mundo"que atacar motivados por el deseo de:

- Jugar(Probar, conocer, entretenerse, todo por curiosidad y por malicia).

[1]CERT: Computer Emergency Response Team, organismo de reporte de incidentes

- Destruir (Denegación de servicios, redirigir las peticiones a otro servidor,etc.).
- Espiar (Tecnologías, información).
- Robar (Información, programas, dinero, bienes en forma activa o pasiva, etc.).

Se pueden adoptar actividades variadas respecto a la protección, entre ellas:

- No hacer nada. Lo que nos otorga una seguridad mínima.
- Considerar todo como una gran caja negra. Bajo la filosofía que en ellas nadie entra. Lo que es evidentemente falso.
- Establcer seguridad a nivel de Host. Lo que requiere muchos esfuerzos y recursos cuando hay muchos usuarios y equipos interconectados.
- Establcer seguridad a nivel de red. Se orienta al control de la red personal, al sistema operativo de la red, más que a la protección de recursos individuales.

Con el uso de mecanismos de seguridad se minimizará el problema persistente de la temática de la seguridad. Para equilibrar el riesgo de las redes con el préstamo de servicios podemos considerar como un entorno seguro a un sistema computacional que cuente generalmente con las siguientes características:

Privacidad. La información puede ser manipulada solo por quien tenga autorización para hacerlo ya sea el dueño o pertenezca al grupo de trabajo que tienen permiso de acceso.

Integridad. La información es consistente, fiable y no contiene alteraciones indeseadas.

Disponibilidad. La información es accesible cuando el usuario lo requiere. Esto compete a las políticas de acceso a la información.

La seguridad informatica está soportada por tres grandes ramos:

Autenticar Se refiere a establecer las entidades a las que se les concede acceso al universo de recursos de cómputo otorgado por algún servidor.

Autorizar Define las entidades autorizadas a tener acceso a los recursos de cómputo y garantiza que tengan efectivamente acceso únicamente a las áreas de trabajo sobre las cuales deben tener dominio.

Auditoría Se refiere a la continua vigilancia de los servicios en producción. Dentro de este rubro, se considera mantener estadísticas de acceso, estadísticas de uso y políticas de acceso físico a los recursos.

Debido a la amplitud del tema a tratar, nos enfocaremos en las siguientes vertientes principales: La auditoría, con un enfoque exclusivo en la detección de intrusiones en equipos conectados a la red; el análisis forense en un equipo atacado para la recuperación parcial o total de sus datos y para obtener una estimación de los datos dañados y su prevención para futuros ataques, al tratar de obtener toda evidencia posible que lleven a encontrar al atacante.

En la fase de análisis forense, se analizarán los equipos atacados intentando reconstruir la secuencia temporal de las actividades realizadas por el atacante, de tal forma que sirva tanto para ayudar a recuperar datos perdidos y regresar al sistema a su estado de funcionamiento normal, así como para obtener pruebas de la actividad delictiva. Esta fase concluirá con el desarrollo de un método que facilite el proceso de análisis forense. A su vez se describirán los pasos a seguir en el análisis de la red, políticas y puesta en marcha de un servidor, cuya finalidad tenderá a minimizar los riesgos de los ataques, por medio de detectores de intrusos bien sea locales o de red.

La tesis, se orienta a la realización de un estudio pormenorizado de lo que ha pasado en algunos `equipos víctima`, a partir de la información que ha quedado registrada en las bitácoras del equipo de control (tráfico de red destinado a los equipos víctima) y, sobre todo, de la información que se pueda recuperar de los equipos atacados. Es decir, se trata de reconstruir la secuencia y la temporalidad de los pasos dados por el atacante, obteniendo la mayor cantidad posible de información.

El resultado del análisis permitirá proponer métodos e incorporarlos como una politica de seguridad, los pasos que se deben seguir cuando un equipo ha sido atacado,de manera que la información obtenida permita prevenir futuros ataques y recuperar la mayor cantidad de datos posibles a fin de que se convierta en evidencia y de ser necesario se podría obtener pruebas legales contra el atacante en caso de decidir llevar el caso ante las leyes.

En México no hay antecedentes de personas hayan sido juzgadas y detenidos por delitos informáticos; la más reciente información que se tiene es acerca de la legislación en materia penal actual en la que ya contempla este tipo de delitos y las sanciones correspondientes, por lo que es necesario iniciar la cultura de seguimiento.

Capítulo 1

Seguridad en sistemas Unix

Definiré el significado explícito de lo que se conoce como seguridad, un entorno general para el sistema operativo Unix [163], la estructura de su sistema de archivos, beneficios, aspectos generales de las redes con protocolos `tcp/ip` y la seguridad que se encuentra en ella. Y para concluir en este capitulo entrelazaré cada punto expuesto para definir el objetivo de seguridad en sistemas Unix.

La seguridad es el mecanismo que garantiza el buen funcionamiento de un sistema. En realidad la seguridad para sistemas de cómputo es un concepto cuya definición exacta es difícil de proporcionar, debido a la gran cantidad de factores que intervienen en computación. Sin embargo, es posible decir que seguridad es el conjunto de recursos (metodologías, documentos, programas y dispositivos físicos) encaminados a lograr que los recursos de cómputo disponibles en un cierto entorno, pueda ser accesible única y exclusivamente por quienes tienen la autorización para hacerlo [114]. Es establecer un control de cualquier función ya sea error interno o externo al sistema de cómputo, esto para evitar accidentes o daños a si mismo a otro residente de la misma red.

En el mercado computacional existe una amplia gama de sistemas operativos [121], y por considerarlo ventajoso un sector de la población está emigrando a Unix [146] o alguna de sus variantes debido a su trayectoria de más de treinta años de desarrollo. Unix cuenta con múltiples servicios de red que contemplan la coexistencia con las redes privadas y públicas.

1.1. Unix

Es un sistema operativo, su estructura básica es distribuida por una gama de versiones y distribuidores. El sistema operativo Unix está pensado para atender a varios usuarios simultáneamente. Desde un servidor central, se pueden conectar las terminales que sean necesarias, con una cantidad de memoria adecuada, en general 1Mb por usuario mediante conexiones al puerto serie del servidor o mediante una Red de Área Local (LAN) y el protocolo TCP/IP:

Además, cada usuario puede estar trabajando con un programa diferente, compartiendo datos o no, con los demás. Es decir, el sistema trabaja con varios programas a la vez, es multiprogramación.

El sistema operativo Unix trabaja así porque dedica espacios cortos de tiempo para atender a cada uno de los usuarios conectados al sistema.

Unix utiliza parte del espacio del disco como memoria virtual[1]. De manera que, los procesos que en determinado momento están ejecutándose están en la memoria central, y van intercambiándose con los demás a medida que va siendo necesario.

Por todo lo expuesto anteriormente el sistema operativo Unix es:

- Multiusuario.
- Multiprogramación.
- Con procesamiento en Tiempo Compartido
- Utiliza Memoria Virtual Constantemente (Swap)

Cada usuario está asignado a trabajar en determinadas zonas, mediante permisos de lectura, escritura y ejecución. En general, un usuario podrá moverse y mirar en todo el sistema, pero sólo podrá ejecutar y modificar en su zona de trabajo.

Existe un usuario especial encargado de Administrar el sistema, es decir, realizar los trabajos de creación de usuarios, dotación de permisos, instalación, etc. Es más conocido como superusuario o root, que tiene acceso a todo el sistema y puede utilizar toda la potencia del mismo.

[1]Memoria Virtual: también conocida como SWAP

Kernel

Es el núcleo del sistema operativo Unix. Tiene diversas tareas asignadas: Planificar, coordinar y gestionar la ejecución de los procesos. Para ello, hace uso de las prioridades asignadas a cada proceso y utiliza algoritmos específicos para repartir el tiempo entre los diversos procesos que compiten por él. Dar servicios del sistema, como entrada/salida y gestión de ficheros. Manejar las operaciones dependientes de hardware, es decir, realiza las funciones de más bajo nivel de manera que se oculten al usuario. Un kernel típico puede constar de unas 20.000 líneas de código de las cuales un 70-80 % está escrito en C y el resto depende de la máquina. Para un PC ocupa unos 500 Kb y para máquinas grandes puede llegar a 2 Mb.

1.1.1. Ventajas

Una ventaja de los sistemas Unix sobre otros sistemas, es que toda la configuración se guarda en archivos de texto que pueden ser modificados por cualquier editor, aunque se el más austero, a diferencia, por ejemplo del desafortunadamente celebre Registro de Windows, una base de datos que, si llega a corromperse significa la absoluta corrupción del sistema completo.

El sistema operativo Unix , en sus inicios solo se instalaba en sistemas robustos como HP/UX, AIX servidores de cientos de miles de dólares. Otros (Solaris, IRIX) funcionan en máquinas medianas y grandes, esto es con supercomputadoras de 32 procesadores o más. Existen otros sistemas diseñados para correr en computadoras pequeñas; servidores basados en CPUs compatibles con Intel (SCO, IBSD). Otra categoría completamente diferente: FreeBSD, NetBSD, OpenBSD y Linux. Éste último otorga la licencia de GNU (Gnu No es Unix); lo que significa que el código es libre y modificable. Con Linux puede haber ventajas y desventajas; cualquiera puede colocar un servidor, pero a su vez puede modificar el código no sólo el administrador o los usuarios, sino también los posibles atacantes, aunque cualquier persona puede tener un servidor conectado a Internet sin la necesidad de pagar mucho por una estación de trabajo. Estos sistemas operativos son sistemas libres. Son sistemas operativos de distribución gratuita, que ayudan a reducir sensiblemente el costo en máquinas de porte pequeño y medio, que brindan la libertad de poder aprender de su código fuente, o bien modificarlo para cumplir alguna necesidad específica y redistribuirlo con los cambios que le hagamos. Esto es lo que ha llevado a que estos sistemas operativos corran prácticamente en cualquier computadora.

Unix fue desarrollado para brindar un entorno de trabajo multitareas (esto significa que varios procesos sean ejecutados al mismo tiempo) y multiusuarios (esto es que pueden estar

en terminales remotas trabajando directamente con las utilerías, compiladores, servicios de red, e-mail, entre otros).

Cuando se habla de Unix no se puede evitar hacer hincapié en las redes, así que como se mencionó, en el desarrollo de Unix se trabajó en un entorno multiusuarios que no se limitó a una localidad; las fronteras se rompieron por la comunicación vía módem como se muestra en la figura 1.1.

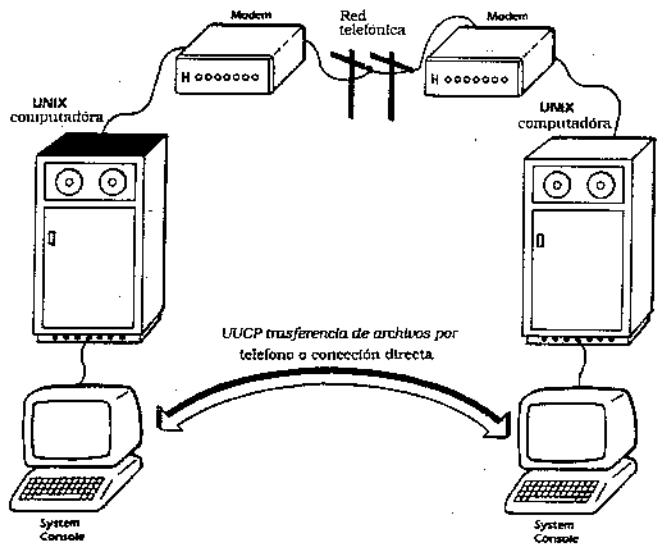


Figura 1.1: Unix, una herramienta no limitada por la distancia

El directorio / (raíz) es el único que no tiene nombre. Cada usuario tiene un directorio home que es el directorio asignado a ese usuario para almacenar sus archivos, estos a través de la variable HOME [163]; A su vez es `/home` o en otras versiones de Unix `/usr/export/home` y `/usr`, donde son asignados las carpetas de los usuarios en caso de que sea servidor de archivo `/var/log /usr/adm /var/adm`, son las bitácoras del sistema.

1.1.2. Interfaz de Usuario

La interfaz de usuario es el medio de comunicación del usuario con el sistema, puede ser convencional(modo texto o gráfico), similar a lo que conoce como windows.

La interfaz de usuario necesita un intérprete de órdenes que sea capaz de traducir las sentencias de ejecución del usuario para que puedan utilizar el sistema.

Shell

El intérprete de comandos u órdenes de Unix se llama Shell, y se encarga de efectuar las llamadas al sistema operativo correspondiente a las órdenes introducidas por los usuarios.

Se agrupan en cuatro grupos o clases básicas:

- Bourne shell o shell standard: Es el intérprete de comandos original. El oficial que se distribuye con los sistemas Unix y el más extendido. Fue desarrollado a principios de los setenta por Stephen R. Bourne en AT&T. Se invoca con la orden "sh".
- Restricted shell: Es un derivado, con las mismas características y órdenes, pero no con todas sus posibilidades. Se invoca con "rsh".
- C shell: Incorpora muchos conceptos del lenguaje C. Es más lento en su ejecución. Está siempre disponible como shell alternativo en todas las versiones de Unix. Fue desarrollado por Berkeley a mediados de los setenta. Se invoca con "csh".
- Uucp-shell: Está destinado a conectar distintos sistemas Unix. Es más un sistema de comunicaciones.

Otros shell's

- ksh shell: Tiene mejores características que los anteriores. Fue desarrollado a principios de los ochenta por David Kron.
- Visual shell: Es un entorno básico de trabajo basado en menús. Es ideal para usuarios no muy expertos.
- System Administrator shell: Es el administrador del sistema, realiza opciones propias del superusuario, pero en un entorno de menús. Se invoca con "sysadmsh".

Shell's propios de Linux

- Bourne: Se invoca con "sh" y lo reconoceremos por el prompt "$".
- Cshell: Utiliza las mismas órdenes que el Bourne, pero con ventajas como llevar un histórico de órdenes y llevar un control de tareas. Se invoca "tcsh".
- Ash: Se invoca así y es un shell reducido. Se utiliza cuando la memoria es muy limitada.
- Korn shell: Es de dominio público y se invoca con "pdksh".
- Bourne Again shell: Es el shell por defecto de Linux, y amplía las capacidades del Bourne. Se invoca con "bash".

El superusuario debe asignar a cada usuario, en el momento de la creación de su cuenta, el shell que éste vaya a utilizar y de esta forma, configurar el entorno de trabajo de acuerdo a ese usuario. También es posible cambiar de shell momentáneamente para después volver al de origen.

Las características más destacables de este shell es la versatilidad, es decir, la facilidad de modificación y adaptación a las necesidades y preferencias de cada usuario.

1.1.3. Manipulación de archivos

El núcleo[2] de Unix interpreta los archivos como secuencias de Bytes. Todos los archivos se manipulan de la misma forma. Unix, lo gestiona todo por medio de archivos, pantalla, impresora, etc.

Unix posee una estructura de directorios de tipo `árbol` jerárquica gestionable de forma muy potente. Los directorios son en realidad archivos que contienen otros archivos o directorios.

Unix utiliza drivers o archivos controladores de dispositivos para representar los diferentes dispositivos periféricos. Es decir, al acceder a estos drivers, en realidad, se está accediendo a los dispositivos que estos representan.

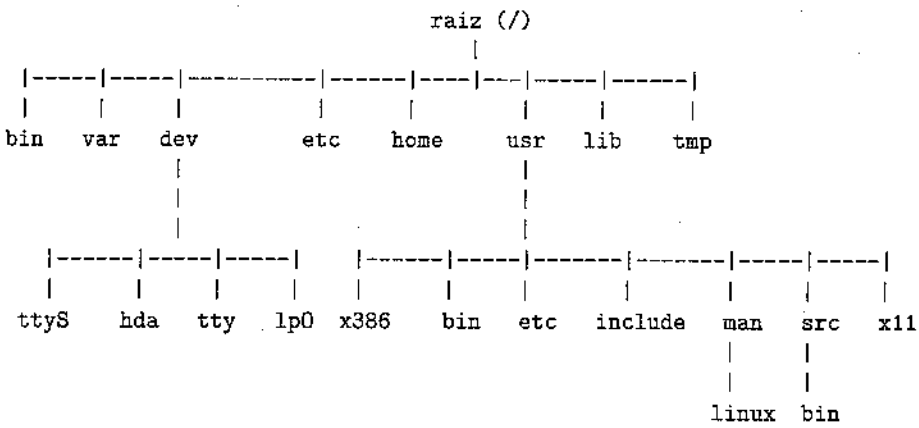
[2]Núcleo: es también llamado Kernel en inglés y es de hecho la parte principal del sistema operativo, la que se ocupa de gestionar los recursos de la memoria, habilitar el acceso a los sistemas de archivo, gestionar diversas de la red, etc.

Tipos de archivos

- Ordinarios: Contienen texto, datos o programas.
- Directorios: archivo que contiene otros archivos o directorios.
- Driver o archivo controlador: archivo que representa un dispositivo físico concreto.

Unix se organiza en sistema de archivos. Se entiende por tal al conjunto de archivos, directorios y la información que Unix utiliza para su gestión.

Unix se organiza en sistema de archivos desde su instalación, en la que crea una serie de directorios necesarios para el buen funcionamiento del sistema. El sistema de archivos inicial de Unix sin usuarios es el que a continuación se detalla. Se debe además mencionar que esta estructura de archivos cambia según la version de Unix que estemos utilizando:



En Unix existe una jerarquía de directorios, que para un sistema estándar sería:

Nomenclatura	Descripción
/	Es la sección raíz o padre del sistema operativo, donde se ubica cualquier partición.
/dev	Archivos especiales de dispositivos.
/lib	Bibliotecas del sistema.
/bin	Archivos binarios o ejecutables.
/etc	Archivos de configuración restringidas al superusuario.
/tmp	Archivos temporales (se borra periódicamente).
/mnt	Sección donde se montan ligas a dispositivos como CD, Floppy y Cintas.
/usr /usr/lib /usr/bin /usr/man /usr/ucb /usr/ucb	Ordenes de ejecución, bibliotecas y programas adicionales.
/sbin	Archivos de ejecución sólo por el administrador
/home	Directorio de usuarios
/proc	Estructura virtual de archivos
/var	Bitacora del sistema
/boot	Información necesaria para el sistema de arranque
/dev/console	Sistema de consola
/dev/ttyS	Acceso a puertos
/dev/cua	Acceso a puertos
/dev/hda	Primer disco duro
/dev/sda	Primer disco duro SCSI
/dev/lpO	Primer puerto paralelo
/dev/tty	Consolas virtuales
/dev/pty	Seudoterminales
/usr/x386	Sistema X-window
/usr/bin	Archivos binarios o ejecutables
/usr/etc	Información y programas
/usr/include	Archivos para compilador C
/usr/man	Páginas Man
/usr/src	Código fuente
/usr/src/linux	Código fuente del núcleo linux
/var/adm	Archivos de administración
/var/spool	Archivos de Spool
/usr/x11/bin	Ejecutables de X-window

1.1.4. Control de procesos

Un proceso es la actividad resultante de la ejecución de un programa. Los procesos pueden ser controlados directamente por los usuarios, mediante las utilidades disponibles en Unix, que son:

- Creación de procesos.
- Eliminación de procesos (matar procesos).
- Información del estado de los procesos.
- Planificación de procesos.
- Control en tiempo real.

1.1.5. Información del Sistema

En Unix existe un conjunto de llamadas al sistema para obtener información:

- De carácter general (hora, fecha, etc).
- Local a los terminales (línea serie, IP Address, etc).
- Local a los usuarios (nombre de los terminales activos, etc).
- Relativo al Sistema (características, contabilización, etc).

1.2. Redes

Las redes de computadoras se comunican por protocolos esto por el intercambio de información entre ellas y por un medio físico, dependiendo de las necesidades y la arquitectura.

Para la comunicación entre dos entidades computacionales situadas en sistemas diferentes, se necesita definir y utilizar un protocolo y sus medios físicos y/o lógicos [86].

1.2.1. Protocolos

Un protocolo [91] es un conjunto de reglas que indican cómo se debe llevar a cabo un intercambio de datos o información. Para que dos o más nodos en una red puedan intercambiar información es necesario que menejen el mismo conjunto de reglas, es decir, un mismo protocolo de comunicaciones.

Debido a la gran variedad de protocolos, se hizo necesario estandarizarlos y para eso se tomó un diseño estructurado o modular que produjo un modelo jerárquico conocido como modelo de referencia OSI (Open Systems Interconnection).

1.2.2. Topologías de red

En la siguiente tabla se muestra brevemente el tipo de cableado, el protocolo como su topología. Para mayor información consulta el anexo de redes [122].

Topología	Cableado	Protocolo
Bus	Coaxial Par Trenzado Fibra óptica	Ethernet LocalTalk
Estrella	Par trenzado Fibra óptica	Ethernet LocalTalk
Estrella en Anillo	Par trenzado	Token Ring
Arbol	Coaxial Par trenzado Fibra óptica	Ethernet

1.2.3. Medios de transmisión

Existen diferentes medios de intercambiar datos entre dos host a través de una o más redes; por su importancia se mencionan los siguientes:

Par trenzado : Es el medio de comunicación más barato y más usado. Consiste en un par de cables, recubriertos con aislante, para cada enlace de comunicación. Debido a que puede haber interferencia entre pares, estos se trenzan con pasos diferentes. La utilización del trenzado tiende a disminuir la interferencia electromagnética.

Este tipo de medio es el más utilizado debido a su bajo costo (se utiliza mucho en telefonía) pero su inconveniente principal es su poca velocidad de transmisión y su corta distancia de alcance. Con estos cables, se pueden transmitir señales analógicas o digitales.

Cable coaxial : Consiste en un cable conductor interno (cilíndrico) separado de otro cable conductor externo por anillos aislantes o por un aislante macizo. Todo esto se recubre por otra capa aislante que es la funda del cable.

Este cable, aunque es más caro que el par trenzado, se puede utilizar para cubrir mayor distancia, con velocidades de transmisión superiores, menos interferencias y permite conectar más enlaces. Se suele utilizar para televisión, telefonía a larga distancia, redes de área local, conexión de periféricos a corta distancia, etc. Se utiliza para transmitir señales analógicas o digitales.

Sus inconvenientes principales son: atenuación, ruido térmico, ruido de intermodulación, dificultad para agregar o eliminar equipo.

Fibra óptica : Se trata de un medio muy flexible y muy fino que conduce energía de naturaleza óptica. Su forma es cilíndrica con tres secciones radiales: núcleo, revestimiento y cubierta. El núcleo está formado por una o varias fibras muy finas de cristal o plástico. Cada fibra está rodeada por su propio revestimiento que es un cristal o plástico con propiedades ópticas distintas a las del núcleo. Alrededor de este conglomerado está la cubierta (constituida de material plástico o similar) que se encarga de aislar el contenido de aplastamientos, humedad, etc. sus principales ventajas son:

- Es un medio muy apropiado para largas distancias e incluso últimamente para LAN's .
- Sus beneficios frente a cables coaxiales y pares trenzados son :
  - Permite mayor ancho de banda.
  - Menor tamaño y peso.
  - Menor atenuación.
  - Aislamiento electromagnético.
  - Mayor separación entre repetidores.

Su rango de frecuencias es todo el espectro visible y parte del infrarrojo.

Transmisión inalámbrica : Se utiliza principalmente el aire. Se manda energía electromagnética por medio de una antena y luego se recibe está energía con otra antena.

Hay dos configuraciones para la emisión y recepción de esta energía: Direccional, toda la energía se concentra en un haz que es emitido en una cierta dirección, por lo que tanto el emisor como el receptor deben estar alineados. En el método unidireccional, la

energía se dispersa en múltiples direcciones, por lo que varias antenas pueden captarla. Cuanto mayor es la frecuencia de la señal a transmitir, más factible es la transmisión unidireccional.

Por tanto, para enlaces punto a punto se suelen utilizar microondas (altas frecuencias). Para enlaces con varios receptores posibles se utilizan las ondas de radio (bajas frecuencias). Los infrarrojos se utilizan para transmisiones a muy corta distancia (en una misma habitación).

1. Microondas terrestres: Se utilizan antenas parabólicas. Para conexiones a larga distancia, se utilizan conexiones intermedias punto a punto entre antenas parabólicas.

   Se suelen utilizar en sustitución del cable coaxial o las fibras ópticas ya que se necesitan menos repetidores y amplificadores, aunque se necesitan antenas alineadas. Se usan para transmisión de televisión y voz.

   La principal causa de pérdidas es la atenuación debido a que las pérdidas aumentan con el cuadrado de la distancia (con cable coaxial y par trenzado son logarítmicas). La atenuación aumenta con las lluvias. Las interferencias son otro inconveniente de las microondas ya que al proliferar estos sistemas, peude haber más solapamientos de señales.

2. Microondas por satélite: El satélite recibe las señales y las amplifica o retransmite en la dirección adecuada. Para mantener la alineación del satélite con los receptores y emisores de la tierra, el satélite debe ser geoestacionario.

   Se suele utilizar este sistema para:

   - Difusión de televisión.
   - Transmisión telefónica a larga distancia.
   - Redes privadas.

   El rango de frecuencias para la recepción del satélite debe ser diferente del rango al que este emite, para que no haya interferencias entre las señales que ascienden y las que descienden.

   Debido a que la señal tarda un intervalo de tiempo pequeño desde que sale del emisor en la Tierra hasta que es devuelta al receptor o receptores, debe de tenerse cuidado con el control de errores y de flujo de la señal.

   Las diferencias entre las ondas de radio y las microondas son:

   - Las microondas son unidireccionales.
   - Las microondas son más sensibles a la atenuación producida por la lluvia.

- En las ondas de radio, al poder reflejarse en el mar u otros objetos, pueden aparecer múltiples señales "hermanas".

3. Infrarrojos: Los emisores y receptores de infrarrojos deben estar alineados o bien estar en línea tras la posible reflexión de rayo en superficies como las paredes. En infrarrojos no existen problemas de seguridad ni de interferencias ya que estos rayos no pueden atravesar los objetos (paredes por ejemplo). Tampoco es necesario permiso para su utilización (en microondas y ondas de radio sí es necesario un permiso para asignar una frecuencia de uso).

1.2.4. Protocolos TCP/IP

La Agencia de Proyectos de Investigación Avanzada del Departamento de Defensa de los Estados Unidos de Norteamérica definieron un conjunto de reglas que establecieron cómo conectar computadoras entre sí para lograr el intercambio de información, soportando incluso desastres mayores en la subred. Fue así como se definió el conjunto de protocolos [91] de TCP/IP (TCP/IP Internet Suite of Protocols). Para los años 80 una gran cantidad de instituciones estaban interesados en conectarse a esta red que se expandió por todo EEUU. La Suite de TCP/IP consta de 4 capas principales que se han convertido en un estándard a nivel mundial.

Las capas del modelo TCP/IP

Las capas de la suite de TCP/IP son menos que las del modelo de referencia OSI, sin embargo son tan robustas que actualmente une a más de 3 millones de nodos en todo el mundo.

La capa inferior, que podemos nombrar como física respecto al modelo OSI, contiene varios estándares del Instituto de Ingenieros Electrónicos y Eléctricos (IEEE en inglés) como son el 802.3 llamado Ethernet que establece las reglas para enviar datos por cable coaxial delgado (10Base2), cable coaxial grueso (10Base5), par trenzado (10Base-T), fibra óptica (10Base-F) y su propio método de acceso, el 802.4 llamado Token Bus que puede usar estos mismos medios pero con un método de acceso diferente, el X.25 y otros estándares denominados genéricamente como 802.X. La siguiente capa cumple, junto con la anteriormente descrita, los niveles del modelo de referencia 1,2 y 3 que es el de red. En esta capa se definió el protocolo IP también conocido como çapa de internet". La responsabilidad de este protocolo es entregar paquetes en los destinos indicados, realizando las operaciones de enrutado apropiadas y la resolución de congestionamientos o caídas de rutas.

La capa de transporte es la siguiente y está implantada por dos protocolos: el Transmission Control Protocol y el User datagram Protocol. El primero es un protocolo confiable (reliable) y orientado a conexiones, lo cual significa que nos ofrece un medio libre de errores para enviar paquetes. El segundo es un protocolo no orientado a conexiones (connectionless) y no es confiable (unreliable). El TCP se prefiere para la transmisión de datos a nivel red de área amplia y el otro para redes de área local.

La última capa definida en la suite de TCP/IP es la de aplicación y en ella se encuentran decenas de aplicaciones ampliamente conocidas actualmente. Las más populares son el protocolo de transferencia de archivos (FTP), el emulador de terminales remotas (Telnet), el servicio de resolución de nombres (Domain Name Service DNS), el WWW, el servicio de correo electrónico (Simple Mail Transfer Protocol SMTP), el servicio de tiempo en la red (Network Time Protocol NTP), el protocolo de transferencia de noticias (Network News Transfer Protocol NNTP) y muchos más.

1.2.5. Servicios IP

Los servicios que proporciona IP a TCP son: Send (envío) y Deliver (entrega). TCP utiliza Send para solicitar el envío de una unidad de datos y Deliver es utilizada por IP para notificar a TCP que una unidad de datos ha llegado. Los campos incluidos en estas dos llamadas son: dirección origen y destino de los datos, identificador de bloque de datos, indicador sobre si está permitida la segmentación del bloque, tipo de servicio, tiempo de vida, longitud de los datos. Algunos campos no son necesarios para Deliver.

El tipo de servicio solicitado puede ser de encaminamiento lo más rápido posible, lo más seguro posible, por prioridad, etc.

1.2.6. Protocolo IP

IP es el principal protocolo de la capa de red. Este protocolo define la unidad básica de transferencia de datos entre el origen y el destino, atravesando toda la red de redes. Además, el software IP es el encargado de elegir la ruta más adecuada por la que los datos serán enviados. Se trata de un sistema de entrega de paquetes (llamados datagramas IP) que tiene las siguientes características:

- Es no orientado a conexión debido a que cada uno de los paquetes puede seguir rutas

distintas entre el origen y el destino. Entonces pueden llegar duplicados o desordenados.

- Es no fiable porque los paquetes pueden perderse, dañarse o llegar retrasados.[ref El protocolo IP está definido en la RFC [145] 791]

1.2.7. Direcciones IP

La dirección de origen y destino en el encabezado IP es una dirección global de Internet de 32 bits. De estos 32 bits, algunos identifican la computador y el resto a la red. Estos campos son variables en extensión para poder ser flexibles al asignar direcciones de red. Hay diferentes tipos de redes que se pueden implantar en la dirección de red. Unas son grandes (con muchas subredes), otras medianas y otras pequeñas.

1.2.8. Seguridad en TCP sobre IP

La seguridad implica que los segmentos no se pierdan y que lleguen en la secuencia correcta. En está capa es complicado asegurar la llegada y la secuencialidad de los segmentos. Para comprender esto, veamos siete aspectos relacionados:

1. Transporte en orden: TCP numera los segmentos con el número de orden de los datos que contiene, es decir, si el primer segmento se numera con un 0 y contiene 1200 bytes, el siguiente segmento se numera como 1200.

2. Estrategia de retransmisión: Se usa una estrategia de confirmaciones positivas para que el receptor informe al emisor de la llegada correcta de un segmento (confirmar el 4, significa confirma todos los anteriores). Cuando no se confirma un segmento antes de que expire un temporizador, se debe retransmitir. Para establecer el temporizador se puede hacer siempre con un valor fijo, pero esto no soluciona el problema cuando hay existencia de condiciones cambiantes de tráfico en la red; la utilización de un temporizador que se adapte a las condiciones de la red también tiene sus inconvenientes.

3. Detección de duplicados: cuando un segmento se pierde, el emisor, al no recibir confirmación envía un duplicado, pero supongamos que lo que ocurrió no fue que se perdió sino que expiró el temporizador o se perdió la confirmación, entonces el segmento al receptor le llegan dos duplicados, por lo que debe de ser capaz de conservar uno y desechar el otro. Un problema a tener en cuenta es que la numeración de los segmentos se debe hacer módulo un número muy grande para que no se numeren dos segmentos

con el mismo número y que ambos estén en la red al mismo tiempo. Un problema adicional es que haya segmentos circulando aún cuando la conexión se haya cerrado , si un instante después se abre otra vez, el receptor podría recibir estos segmentos que ya no son válidos y confundirlos con los nuevos de la nueva transmisión, y para solucionar esto, el receptor debe recordar los últimos segmentos que recibió en la última conexión.

4. Control de flujo: el tipo de control de flujo más robusto es el de créditos. Este sistema consiste en que cuando el receptor recibe un segmento, en la confirmación se incluye este segmento y todos los anteriores y además se le indica al emisor que hay disponibilidad para aceptar un número determinado de nuevos segmentos (crédito). Este sistema hace que si se pierde una confirmación, la siguiente confirma a la anteriormente perdida y además, cuando un temporizador del emisor expira, éste volverá a enviar el segmento.

5. Establecimiento de la conexión: se requiere un diálogo entre los dos sistemas para establecer la comunicación y para eso se utiliza una señal de sincronización. Hay un mecanismo para repetir señales de sincronización en caso de que éstas no lleguen. Para evitar confusión en la repetición de señales de sincronización, estas son numeradas, y además tienen un campo de confirmación de haber sido recibidas.

6. Cierre de la conexión: puede darse la situación en que una señal de fin de conexión se anticipe a uno o varios segmentos de datos, entonces, se perderán estos segmentos; para evitar está situación se añade un campo de último segmento a transmitir en el segmento de señalización de final de transmisión, de este modo el receptor esperará los segmentos restantes.

7. Recuperación de caídas: Puede ocurrir que uno de los sistemas falle, caso en el cual se desconectará, perdiéndose todos los datos que se contenían en su configuración. Pero el otro sistema conectado ignora que exista este problema, así que continuará enviando datos hasta que sus temporizadores terminen. Entonces se dará por concluida la desconexión.

1.3. Problemática actual

International Data Corp. (IDC) señaló que a finales de 2002 más de 600 millones de personas, cerca del 10 % de la población mundial, tendrán acceso a Internet. El día 1 de noviembre de 1988 Internet fue " infectada" con un virus de tipo " gusano".

El 10 % de todos los servidores conectados fueron afectados. El acontecimiento subrayó la falta de mecanismos adecuados de seguridad en Internet, por lo que DARPA formó el Com-

puter Emergency Reponse Team (CERT), un equipo de reacción rápida que mantiene datos sobre todas las incidencias en red y sobre las principales amenazas. En 1989 el número de servidores conectados a Internet alcanzaba ya los 100 000 servidores.

El planteamiento inicial de Internet en 1973 y 1974 contempló un total de 256 redes interconectadas. No se contemplaba la posibilidad de que participaran más de esas 256 redes. Cuando las redes locales aparecieron en gran número, se inventó una manera de subdividir las direcciones IP con el fin de mantener el diseño original. Luego se crearon las direcciones de clases A B y C para permitir la conexión de millones de IPs. Por qué es necesario existen propuestas de diferente índole para tratar el problema. Una de ellas y que ya se está implentando es la versión actualizada de TCP/IP. La transición de la versión 4 (actual) de IP a la siguiente generación (versión 6) está siendo investigada para un mejor uso y seguridad en el empaquetamiento.

Algunas de las ventajas de la nueva generación de IP incluyen:

- Aumento de direcciones de 32 bits a 128 bits.
- Encabezados de mensajes simplificadas.
- Encabezados extendidos opcionales que permiten mayor control de seguridad.

Las entes claves en esta transición son los vendedores de equipos de direccionamiento " routers" y software de los sistemas, y los proveedores de servicios Internet. Estos últimos sobre todo, tienen un interés en asegurar que el mayor número de usuarios pueda acceder a sus servicios, sin perder conectividad en el proceso de transición.

Para el mejor manejo de los nombres en Internet se crearon organismos que se encargarán de los servidores de nombres por regiones. Las regiones eran previamente preestablecidas a nivel mundial y a su vez cada país estaría seccionado en organizaciones. Por dar un ejemplo la UNAM pertenece a la red NICUNAM que está regulada por NIC (Network Information Center) México, organización que se encarga de asignar a las direcciones IP su nombre, para facilitar la localización IP de las máquinas conectadas a Internet. El NOC (Network Operation Center), se encarga del enrutamiento o soporte de las redes en cuestión física.

1.4. Valor de la Información

Definiré los valores que puede alcanzar la información y sus características para proteger. Establecer el valor de la información es totalmente relativo; constituye un recurso que, en muchos casos no se valora adecuadamente debido a su intangibilidad(de la información, por ejemplo los correos electronicos, donde se puede cuestionar cuales son valiosos y cuales no y hasta que tiempo lo son), este suceso no pasa con los equipos de cómputo, las aplicaciones y la documentación.

La información se divide en:

Pública: Es la información que puede ser visualizada por cualquier persona. Es la información que esta intensionalmente expuesta.

Privada: Es la información, que sólo puede ser vista por un grupo selecto de personas que trabaja con ella(por ejemplo antecedentes médicos).

Para la información privada, se debera de subdividirla, por ejmplo, propongo las siguintes caracteristicas.

1. Es Crítica: es indispensable para garantizar la continuidad operativa.
2. Es Valiosa: es un activo con valor en sí misma.
3. Es Sensitiva: debe ser conocida por las personas que la procesan y sólo por ellas.

La integridad de la información es la característica que hace que su contenido permanezca inalterado a menos que sea modificado por personal autorizado.

La diponibilidad u operatividad de la información es su capacidad de estar siempre disponible para ser procesada por las personas autorizadas. Esto requiere que la misma se mantemga correctamente almacenada con el hardware y software funcionando perfectamente y que se respeten los formatos para su recuperación en forma satisfactoria.

La privacidad o confidencialidad de la información es la necesidad de que la misma sólo sea conocida por personas autorizadas. En casos de falta de confidencialidad, la información puede provocar severos daños a su dueño.

El control sobre la informción permite asegurar que sólo los usuarios autorizados puede decidir cuando y como permitir el acceso a la misma.

La autenticación permite definir que la información requerida es válida y utilizable en tiempo, forma y distribución. Está propiedad también permite asegurar el origen de la información, validando al emisor de la misma, para evitar suplementación de identidades.

Metodos de autenticación:

- Protección a la réplica: mediante la cual se asegura que una transacción sólo puede realizarse una vez, a menos que se especifique lo contrario. No se deberá poder grabar una transacción para luego reproducirla, con el propósito de copiar la transacción para que parezca que se recibieron múltiples peticiones del mismo remitente original.
- No repudio: mediante lo cual se evita que cualquier entidad que envió o recibió información alegue, ante terceros, que no la envió o recibió.
- Consistencia: se debe poder asegurar que el sistema se comporte como se supone que debe hacer ante los usuario que corresponda.
- Aislamiento: este aspecto, íntimamente relacionado con la confidencialidad, permite regular el acceso al sistema, impidiendo que personas no autorizadas hagan uso del mismo.
- Auditoria: es la capacidad de determinar qué acciones o procesos se están llevando a cabo en el sistema, así como quién y cuando las realiza.

1.5. Seguridad

La seguridad en Internet es uno de los temas más relevantes por la demanda del mercado virtual que implica transacciones financieras, sin menospreciar también otros propósitos como la producción en línea, educación, servicios, integridad, etc. El problema de seguridad en Internet surge porque inicialmente Internet fue creada para el libre acceso a la información y es regida principalmente por las políticas de buen uso de la red.

A partir de 1990, comenzó la preocupación por la seguridad, debido a que la sociedad comercial encontró en Internet un canal de flujo hecho a la medida: rápido, barato y cada vez más extendido y eficiente. Desde entonces grandes compañías de noticias difunden el crecimiento, los beneficios y los peligros del uso de Internet. Las empresas basan sus comunicaciones externas en ella y cada día son más las operaciones financieras que se hacen a través de Internet, que poco a poco está reemplazan de los medios de comunicación tradicionales.

Lógicamente, conforme más información está disponible en Internet, más importancia cobra la protección de dicha información y el control del acceso a la misma. Dentro de este panorama, nos enfrentamos a una creciente realidad, la necesidad de seguridad en los datos, los servicios, las transacciones y las partes involucradas. Las redes y servicios tienen multiples vulnerabilidades y la única solución es minimizar los riesgos. Los sistemas operativos incluyen rutinariamente configuraciones inseguras y con dispositivos de seguridad incompatibles, debido a defectos de producción que dan su correspondiente dosis de agujeros. Además, tampoco se considera excepcional que las aplicaciones fallen, ni encontrar organizaciones donde dichos asuntos ni siquiera sean de preocupación u ocupación de alguien en particular.

La calidad del sistema y su funcionamiento operacional correcto e ininterrumpido definitivamente impactan en forma positiva sobre los beneficios que obtienen las organizaciones.

Objetivo de la seguridad en cómputo

Es garantizar la privacidad de la información y la continuidad del servicio, tratar de minimizar la vulnerabilidad de los sistemas o de la información contenida en ellos, es proteger las redes privadas y sus recursos mientras que se mantienen los beneficios de la conexión a una red pública.

La seguridad es un factor clave en cualquier implementación de red tanto en la actualidad como en el futuro. La seguridad en la red y en Internet tendrán un papel fundamental en las operaciones comerciales ya que las compañías de todo el mundo confían cada vez más en las redes basados en IP para implementar sus estrategias de comercio electrónico. Las compañías ponen mucho en juego, por eso las redes seguras son fundamentales [78].

Origen de la inseguridad

Un factor decisivo es el crecimiento acelerado de las redes empresariales y de particulares. Las redes son inseguras, un parte, debido al Internet, aunado a que el diseño de las redes se asumía en ambientes seguros y controlados a través de usuarios autorizados sin vislumbrar la futura conexión a redes externas. Además los protocolos de comunicación como el TCP/IP no fueron concebidos teniendo en cuenta aspectos concernientes a la seguridad. Estas son las principales causas de la inseguridad en las redes.

En comercio electrónico se cuestiona si las transacciones que se realizan vía Internet son realmente seguras por su autenticación o control de acceso de socio. En otros casos se cues-

tiona si un correo puede ser falsificado por cuestiones políticas utilizando suplantación. Sin precauciones de seguridad en estas transacciones muchas personas podrían pasearse por los datos que estamos transmitiendo, entrar en una conversación o llegar a obtener nuestros datos confidenciales.

La seguridad es algo que se debe tratar en forma explícita e implícito para cualquier persona que desea utilizar a las redes de está nueva sociedad, no sólo afecta al gran empresario sino también al usuario de Internet para leer su correo, cuando un cliente compra un artículo etc.

Acciones y consecuencias

Concretamente, en una comunicación se puede encontrar tres problemas claramente diferenciados:

ESCUCHA DE RED (sniffer) La información no sufre alteración pero usuarios no autorizados pueden acceder a la información en un tiempo relativamente corto.

MODIFICACIÓN La información es modificada, no es necesario que existan los escuchas, el hecho es alterar un documento, por ejemplo, alguien podría cambiar la cantidad a pagar en un pedido que se transmite por la red, sin descartar el acto de modificación en un FTP-anonimo[3].

SUPLANTACIÓN Este problema aparece cuando alguien dice ser quien realmente no es, haciendo posible que una entidad se haga pasar por otra, realizaría ventas que no llegaría a entregar y cobraría sus importes. No es necesariamente forzoso que ingrese a la cuenta de uno de sus servidores, no sólo pasa utilizando cuentas sino también en identidad del servidor.

Para minimizar la problemática se implementan tres aspectos:

CONFIDENCIALIDAD Es la propiedad por la que el destinatario de una comunicación puede conocer la información que se le envía mientras que las personas que no son los receptores no pueden determinar el contenido de lo que les envían.

[3]FTP: Un servicio que hasta la fecha continua, siendo considerado peligroso, pero insustituible, si está correctamente administrado, los permisos de carpetas y archivos pueden ser muy útiles

INTEGRIDAD . Es la propiedad de asegurar que la información sea transmitida desde su origen hasta su destino sin sufrir ninguna alteración o que sufra alteraciones.

AUTENTICACIÓN Es la propiedad de conocer que la información recibida es la misma que la información enviada y quien dice ser el que los envió, realmente los envió.

La información que circula, se procesa y se almacena en una red; se ve sometida a varios tipos de amenazas, como espionaje o acceso no autorizado, copia, alteración, interrupción y destrucción de información o servicios.

1.6. Niveles de seguridad

Existe un patrón de seguridad creado por el gobierno de los Estados Unidos para el uso de las computadoras. Cada empresa o instituto deberá decidir qué tan valiosa es su información. Esto queda estipulado en un libro que puede ser consultado en el sitio [77].

Libro naranja

El libro naranja, hace mención de que tan **confiable** es un sistema más qué decir que tan seguro es un sistema. Recopila estándares de seguridad y lineamientos para el tipo de necesidades fue realizada y analizada por el gobierno de Estados Unidos de America. Sus principales objetivos son [77]:

Medición: Da la métrica para el grado de fiablidad entre los sistemas como por ejemplo es más seguro B2 que un sistema C2[4]:

Guía: Brinda una directriz para el desarrollo y compra de productos fiables.

Adquisición: Proporciona una serie de requisitos de seguridad para las especificaciones informáticas.

El concepto fundamental del libro es medir que tan confiable es un sistema o sitio y certificarlo respecto a un conjunto de criterios de seguridad, por ejemplo:

[4] B2 y C2 son niveles de seguridad, donde B2 tiene en número mayor de implementaciones para considerarlo con mayor seguridad, que se explicarán en los siguientes párrafos.

Políticas de seguridad [135]: Este documento se basa en reglas y prácticas, que un organismo hace y las práctica para proteger la meta en común, en la cual existe un *sujeto* (entidad activa al sistema que puede ser un usuario o un proceso computacional), un *objeto* o parte pasiva, que es manipulada por el sujeto, por ejemplo un archivo o dispositivo, y esto deberá tener una referencia en las **políticas de seguridad técnica** que es la parte que interactúa con el usuario.

Confiabilidad : Es la garantía al depositar en un sistema determinada información y las vías confiables que se han desarrollado, probado, documentado, mantenido y entregado al cliente.

Las especificaciones estandar definen siete niveles de seguridad, denominados A1, B3, B2, B1, C2, C1, D1, siendo el D1 el de menor seguridad y A1 el de mayor. Cada nivel incluye las exigencias de los niveles inferiores a él. Éstas especificaciones se refieren a la autenticación del usuario, la confiabilidad del software de sistema operativo y aplicaciones de usuario.

Nivel D: Estos sistemas tienen exigencias de seguridad mínimos (el sistema entero no es confiable). No se les exige nada en particular para ser considerados de clase D1. El sistema operativo se ve comprometida fácilmente y no existe autenticación respecto de los usuarios y sus derechos a tener acceso a la información almacenada en la computadora. Por lo general se refiere a los sistemas operativos como MS-DOS, MS Windows y el Sistema 7.x de Apple Macintosh.

Nivel C1: Sistema de protección de Seguridad Discrecional. Para que un sistema sea considerado C1 deberá permitir la separación entre datos y usuarios. Al limitar el acceso a determinados datos, los usuarios tienen que identificarse y validarse para ser admitidos en el sistema mediante un identificador único y contraseña que sólo él conoce. Se emplea esta combinación para determinar los derechos de acceso a programas e información que tiene cada usuario. Estos derechos de acceso son los permisos de archivo y de directorio. Los controles de acceso discrecional permiten al dueño del archivo o directorio, y al administrador del sistema, evitar que ciertas personas o grupos tengan acceso a dichos programas o información. Sin embargo, no se impide que la cuenta del administrador del sistema realice alguna actividad. En consecuencia, un administrador poco escrupuloso puede comprometer fácilmente la seguridad del sistema sin que nadie lo sepa. Además, muchas de las tareas cotidianas de administración del sistema sólo pueden ser realizadas por el identificador del usuario llamado raíz (root). Con la actual descentralización de los sistemas de cómputo no es raro que en una organización encontremos dos o tres personas que conocen la contraseña raíz. Esto en sí es un problema, pues no hay forma de distinguir entre los cambios que hicieron ayer `Intruso` o `Victima`. Un sistema típico es el sistema Unix IBM MVS/RACF.

Nivel C2 : Cuenta con características adicionales que crean un ambiente de acceso controlado: debe llevar una *auditoría* de accesos, e intentos fallidos de acceso a objetos (archivos, etcetera). Tiene la capacidad de restringir aún más el que los usuarios ejecuten ciertos comandos o tengan acceso a ciertos archivos, permitir o denegar datos a usuarios en concreto, con base no sólo en los permisos, sino también en los niveles de autorización. Requiere la auditoría del sistema, lo que implica registrar una *auditoría* por cada acción que ocurra en el sistema. La *auditoría* se utiliza para llevar registros de todas las acciones relacionadas con la seguridad, como las actividades efectuadas por el administrador del sistema. La *auditoría* requiere de autenticación adicional para estar seguros de que la persona que ejecuta el comando es quien dice ser y también especifica que los procesos no dejen residuos (datos dejados en registros, memoria o disco por un proceso al abortar su ejecución). La desventaja de la *auditoría* es que requiere de recursos adicionales del procesador y del subsistema de disco. Los usuarios de un sistema C2 tienen la autorización para realizar tareas de administración del sistema sin necesidad de la contraseña raíz. C2 permite llevar mejor la cuenta de las tareas relacionadas con la administración del sistema, ya que cada usuario es quien ejecuta el trabajo y no el administrador del sistema. Ejemplos de sistemas que cumplen está norma son , Computer Associates international: ACF/2/MVS, Digital Equipment Corporation con VAX/VMS 4.3, y HP con MPE V/E.

Nivel B1 : Llamado también Protección de Seguridad Etiquetada, es el primer nivel con soporte para seguridad de multinivel. Se establece que el dueño del archivo no puede modificar los permisos de un objeto que está bajo control de acceso obligatorio. A cada objeto del sistema (usuario o dato) se le asigna una etiqueta, con un nivel de seguridad jerárquico (secreto maximo, secreto absoluto, reservado, etc.) y con unas categorías (contabilidad, nóminas, ventas, etc.). Quienes cumplen con protección obligatoria en la división B1 son: AT&T System V/MLS, UNISIS OS 1100, SecurityWare con el equipo CMW+, IBM MVS/ESA.

Nivel B2 : Conocido como Protección Estructurada. Requiere que todos los objetos estén etiquetados. Los dispositivos como discos cintas y terminales, pueden tener asignado uno o varios niveles de seguridad. Se aborda el problema de la comunicación de un objeto con otro que se encuentra en un nivel de seguridad inferior. Debe tener un modelo teórico de seguridad verificable, debe existir un usuario con los privilegios necesarios para implementar las políticas de control, y éste tiene que ser distinto del administrador del sistema (encargado del funcionamiento general del sistema). Los canales de entrada y salida de datos deben estar restringidos, para evitar fugas de datos o la introducción de estos al sistema. Con este estandar cumplen: Honeywell Information System: con Multics, Trusted Information System XENIX.

Nivel B3 : Llamado Dominio de Seguridad. Requiere un argumento convincente de que el

sistema es seguro. Se utiliza hardware de manejo de memoria para proteger el dominio de seguridad contra accesos no autorizados y modificaciones de objetos (usuario o dato) y definir la protección para cada uno en diferentes dominios de seguridad. Tiene que existir un "monitor de referencia" que reciba las peticiones de acceso de cada usuario y en su caso permita o deniegue el servicio según las políticas de acceso que se hayan definido previamente. El sistema debe ser muy resistente a la penetración de intrusos, así como tener una auditoria que permita detectar posibles violaciones de la seguridad. El unico que proporciona este nivel es Honeywell Federal System XTS-200.

Nivel A1 : Conocido como Diseño Verificado. Cuenta con un proceso estricto de diseño, control y verificación. Para alcanzar este nivel, deben incluirse todos los componentes de los niveles inferiores; el diseño debe verificarse matemáticamente y debe realizarse un análisis de los canales cubiertos y de distribución confiable. Ésto significa que el hardware y el software estén protegidos durante su traslado para evitar violaciones de los sistemas de seguridad. Honey Information System SCOMP y Boeing Aerospace:SNS, estan ubicados como los únicos que cumplen con la protección de verificación.

Pero ¿Qué nivel de seguridad necesita nuestra organización? Esto depende del valor de nuestra información y los recursos que la organización desea invertir para proteger sus recursos y la información.

Si necesita una copia del libro naranja lo puede descargar de:

`http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html`.

1.7. Políticas de seguridad

Son los documentos que describen, la forma adecuada de uso de los recursos de un sistema de cómputo, las responsabilidades y derechos tanto de usuarios como administradores, describe lo que se va a proteger y lo que se está tratando de proteger, éstos documentos son el primer paso en la construcción de *Firewalls* efectivos. Las políticas son parte fundamental de cualquier esquema de seguridad eficiente y con esto la administracón controla sus recursos para obtener mejores resultados, como por ejemplo si se tiene en correcto análisis de red el manejo de los paquetes y se evitan ruteo inecesario. Esto hace más rapida y más segura a la red.

Políticas de seguridad de cómputo

La forma adecuada para plantear la planificación de la seguridad en una organización debe partir siempre de la definición de una política de seguridad que defina el qué se quiere hacer en materia de seguridad en la organización para a partir de ella decidir mediante un adecuado plan de implementación el cómo se alcanzarán en la practica los objetivos fijados.

La Política de Seguridad deberá englobar los objetivos, conductas, normas y métodos de actuación y distribución de responsabilidades y actuará como documento de requisitos para la implementación de los mecanismos de seguridad. Debe contemplar al menos la definición de funciones de seguridad, la realización de un análisis de riesgos, la definición de normativa y procedimientos, la definición de planes de contingencia ante desastres y la definición del plan de auditoría.

A partir de la Política de Seguridad se podrá definir el Plan de Implementación, que es muy dependiente de las decisiones tomadas en ella, en el que se contemplará: el estudio de soluciones, la selección de herramientas, la asignación de recursos y el estudio de viabilidad.

Hay dos cuestiones fundamentales que deberán tenerse en cuenta para implantar con éxito una política de seguridad: es necesario que la política sea aprobada para que esté respaldada por la autoridad necesaria que asegure su cumplimiento y la asignación de recursos; y es necesario que se realicen revisiones periódicas que la mantengan siempre actualizada y acorde con la situación real del entorno.

La Política de Seguridad y el Plan de Implementación (y la implantación propiamente dicha) están íntimamente relacionados con:

- La Política de Seguridad que define el Plan de Implementación ya que la implementación debe ser un fiel reflejo de los procedimientos y normas establecidos en la política.
- El Plan de Seguridad debe estar revisado para adaptarse a las nuevas necesidades del entorno, los servicios que vayan surgiendo y a las aportaciones que usuarios, administradores, etc. vayan proponiendo en función de su experiencia. La revisión es esencial para evitar la obsolescencia de la política debido al propio crecimiento y evolución de la organización. Los plazos de revisión deben fijarse y permitir además revisiones extraordinarias en función de determinados eventos (por ejemplo, el indice del incidentes).
- El Plan de Implementación debe ser auditado para asegurar que las politica implementada, aceptada y puesta en marcha.

- El Plan de Implementación debe realimentar a la Política de Seguridad. La experiencia, los problemas de implantación, las limitaciones y los avances tecnológicos, etc. permitirán que la política pueda adecuarse a la realidad, evitando la inoperancia por ser demasiado utópica y la mejora cuando el progreso lo permita.

En su estructura debera de contemplarse Análisis de riesgos. Sanciones, uso ético de recursos de cómputo, manejo de incidentes.

El uso de las políticas de cómputo catalogada bajo 2 enfoques:

Permisivo: "Todo lo que no esté explícitamente prohibido está permitido"

Paranoico: "Todo lo que no esté explícitamente permitido está prohibido"

La politica de seguridad debe influir en la planeación de un servidor, el espacio en disco duro[5], la versión del sistema operativo a instalar, las actualizaciones que requiere, el uso de los recursos como por ejemplo memoria RAM a que procesos, etc.

Las políticas tiene un ciclo de vida que va desde su preparación, redacción, edición, aprobación, difusión, revisión, aplicación y actualización.

1.8. Análisis de riesgos

Como extensión de las políticas de seguridad, debemos analizar la información qué están en riesgo. No debemos descuidar a un sistema con información altamente sensible o delicada tan sólo por prestarle igual atención que a uno de menor importancia. Es claro que no existe ningún servidor que no requiera atención. Cualquier intrusión en nuestra red, por más insignificate que sea en un servidor, representa una incursión exitosa en nuestro terreno, y la aparición de un sistema base (que creemos seguro) desde el cual puede originarse un ataque a nuestros otros servidores [105].

Las PSC se pueden dividir en varios incisos:

- Seguridad Física - ¿Quién tiene acceso a nuestros sistemas? ¿Cómo están protegidos

[5]Disco Duro(HDD) Área de almacenamiento físico

nuestros equipos de eventos tales como temblores o incendios? ¿Qué tan confiable es la alimentación eléctrica?

- Seguridad de la información - ¿Tenemos respaldos diarios de la información? ¿Están en un medio confiable y en un lugar adecuado?¿Alguien más tiene copia del respaldo? ¿Un usuario debe tener acceso de lectura a los archivos de otro? ¿Y de escritura?

- Seguridad ante ataques - ¿Tenemos procedimientos escritos para actuar en caso de una emergencia informática? ¿Conocemos perfectamente que se ejecutan en nuestro sistema? ¿Podemos identificar - ya sea manual o por medio de una herramienta - si hay alguna modificación en el sistema? ¿Tenemos contacto y relación con un Equipo de Respuestas a Incidentes en Computación?

Para asegurar que se consideran todas las posibles eventualidades, se deberá de elaborar una lista de todos los riesgos conocidos, para lo cual se deberá realizar un análisis de riesgos.

El análisis de riesgos supone más que el hecho de calcular la posibilidad de que ocurran cosas negativas. Se puede obtener una evaluación económica del impacto de estos sucesos negativos. Este valor se podrá utilizar para contrastar el costo de la protección de la Información en análisis y la versión del costo de volverla a producir (reproducir).

La evaluación de riesgos y presentación de respuestas debe prepararse de forma personalizada para cada organización. La evaluación de riesgos supone imaginarse lo que puede ir mal y a continuación estimar el costo que supondría. Se deberá tener en cuenta la probabilidad de que sucedan cada uno de los problemas posibles. De está forma se pueden priorizar los problemas y su costo potencial, desarrollando un plan de acción adecuado.

El análisis de riesgos supone responder a preguntas del tipo:

¿Qué puede ir mal?

¿Con qué frecuencia puede ocurrir?

¿Cuáles serían sus consecuencias?

¿Qué confiables con las respuestas a las tres primeras preguntas?

En lo fundamental la evaluación de riesgos que se lleve a cabo ha de contestar, lo más fiable posible, a las siguientes preguntas:

¿Qué se intenta proteger?

¿Cuál es su valor para un miembro o para toda la organización?

¿Cuál es la probabilidad de que ocurra un ataque?

A continúación se muestra un ejemplo de cómo se realiza una evaluación de riesgos.

El o los responsables de la oficina de informática se sentarán con los responsables de las áreas usuarias y realizarán el siguiente conjunto de puntualizaciones:

¿A qué riesgos en materia de seguridad informática se enfrenta la Institución?

Al fuego, que puede destruir los equipos y archivos.

Al robo común, llevándose los equipos y archivos.

Al vandalismo, que daña los equipos y archivos.

A fallas en los equipos, que dañen los archivos.

A equivocaciones, que dañen los archivos.

A la acción de virus, que dañen los equipos y archivos.

A terremotos, que destruyen el equipo y los archivos.

A accesos no autorizados, filtrándose datos no autorizados.

Al robo de datos, difundiéndose los datos sin cobrarlos.

Esta lista de riesgos que se puede enfrentar en la seguridad, es bastante corta. La Institución deberá profundizar en el tema para poder tomar todas las medidas del caso. Luego de elaborar está lista, el personal de la Institución estará listo para responder a los efectos que estos riesgos tendrán para su Institución.

- ¿Qué probabilidad hay de que tenga efecto alguno de los riesgos mencionados?

Al fuego, que puede destruir los equipos y los archivos

¿La Institución cuenta con protección contra-incendios?

¿Se cuenta con sistemas de aspersión automática?

¿Se cuentan con extintores?

¿Detectores de humo?

¿Los empleados están preparados para enfrentar un posible incendio.?

- A un robo común, llevándose los equipos y archivos

¿En que tipo de vecindario se encuentra la Institución?

¿Las computadoras se ven desde la calle?

¿Existe personal de seguridad en la Institución?

¿Con cuántos vigilantes contamos?

¿Los vigilantes, están ubicados en zonas estratégicas?

- Al vandalismo, que daño los equipos y archivos

¿Existe la posibilidad que un ladrón desilusionado o frustrado cause daños?

¿Hay la probabilidad de qué causen algún otro tipo de daño intencionado?

- A fallas en los equipos, que dañen los archivos

¿Los equipos tienen un mantenimiento continuo por parte de personal calificado?

¿Cuáles son las condiciones actuales del hardware?

¿Es posible predecir las fallas a que están expuestos los equipos?

A equivocaciones que dañen los archivos

¿Cuánto saben los empleados de computadoras o redes?

Los que no conocen del manejo de la computadora, ¿saben a quién pedir ayuda?

Durante el tiempo de vacaciones de los empleados, ¿qué tipo de personal los sustituye y qué tanto saben del manejo de computadoras?

- A la acción de virus, que daño los archivos

¿Se instala software en la oficina sin hacerle un examen previo?

¿Está permitido el uso de disquetes en la oficina?

¿Todas las máquinas tienen unidades de disquetes?

¿Se cuentan con procedimientos contra los virus?

Las computadoras cuentan con modem o salidas permitidas al exterior saliendo de la infraestructura de la red. Y que mecanismos de protección implementan

- A terremotos, que destruyen los equipos y archivos

¿La Institución se encuentra en una zona sísmica?

¿El edificio cumple con las normas antisísmicas?

Un terremoto, ¿cuánto daño podría causar?

- A accesos no autorizados, filtrándose datos importantes

¿Cuánta empresas se dedican a la misma área de investigación o desarrollo similar a la de nuestra organización e infiltran espias, es decir competencia desleal?

¿Qué probabilidad hay que un competidor intente hacer un acceso no autorizado?

¿El modem se usa para llamar fuera y también se puede utilizar para comunicarse hacia dentro?

¿Contamos con Sistemas de Seguridad en el Correo Electrónico o Internet?

- Al robo de datos; difundiéndose los datos.

- ¿Cuánto valor tienen actualmente las Bases de Datos?
- ¿Que pérdida podría causar en caso de que se hicieran públicas?
- ¿Se ha elaborado una lista de los posibles sospechosos que pudieran efectuar el robo?
- La lista de sospechosos, ¿es amplia o corta?
- El fraude, con desvío de fondos.

¿Cuántas personas se ocupan de la contabilidad de la Institución?

¿El sistema de contabilidad es confiable?

Las personas que trabajan en el departamento de contabilidad, ¿qué tipo de antecedentes laborales tienen?

¿Existe acceso al Sistema Contable desde otros Sistemas o Personas?

Para cada riesgo, se debe determinar la probabilidad de factor de riesgo. Como ejemplo se mencionan algunos factores de riesgo y pongo en incapie que cada caso de implementación es único:

Factor de riesgo bajo

Factor de riesgo muy bajo

Factor de riesgo alto

Factor de riesgo muy alto

Factor de riesgo medio

ANÁLISIS DE FALLAS EN LA SEGURIDAD

Este análisis supone estudiar las computadoras, su software, localización y utilización con el objeto de identificar los resquicios en seguridad que pudieran suponer un peligro. Por ejemplo, si se instala una computadora personal nueva, para recibir informes de inventario desde otras PC's vía modem situados en lugares remotos, y debido a que el modem se ha de configurar para que pueda recibir datos, se abre una vía de acceso al sistema informático.

Habrá que tomar medidas de seguridad para protegerlo, un ejemplo puede ser la validación de la clave de acceso.

PROTECCIONES ACTUALES Son los aspectos a considerar, de protección minima [78], donde el objetivo principal es:

Generales: Se hace una copia casi diaria de los archivos que son vitales para la Institución.

Robo común: Se cierran las puertas de entrada y ventanas.

Vandalismo: Se cierra la puerta de entrada.

Falla de los equipos: Se tratan con cuidado, se realiza el mantenimiento de forma regular, no se permite fumar, deberá está previsto el préstamo de otros equipos.

Daño por virus: Todo el software que llega deberá analizarse en un sistema utilizando software antivirus. Los programas de dominio público y de uso compartido (Shareware), sólo se usan si proceden de una fuente fiable.

Equivocaciones: Los empleados tienen buena formación. Cuando sea necesario, se intentan emplear a buenos trabajadores temporales.

Terremoto: Si la construcción y la protección contra incendios, es buena.

Acceso no autorizado: Se cierra la puerta de entrada. Varias computadoras disponen de llave de bloqueo del teclado.

Fuego: Si se encuentra instalado un sistemas contra incendios, ya sea si contamos con extinguidores, en sitios estratégicos, se deberá pretender brindar entrenamiento en el manejo de los sistemas de incendio o en su caso de los extinguidores, siendo ideal que fuera en forma periódica.

1.9. Detección

Significa percibir una acción indebida ante la observación minuciosa de un error [97], donde el error puede ser debido a una falta de atención, o bien la ruptura de propiedades no otorgadas.

Detectar intrusiones es conocer el acto de penetración al servidor o servicio ante ciertas características. Se determina a partir de anomalidades del comportamiento y del uso que hacen los atacantes de los recursos del sistema. Este tipo de detección pretende cuantificar

el comportamiento normal de un usuario o de las redes. Para una correcta distinción hay que tener en cuenta las tres distintas posibilidades que existen en un ataque, atendiendo a quién es quien lo lleva a cabo:

Penetración externa. Es la intrusión que se lleva a cabo a partir de un usuario o un sistema de computadoras no autorizado, para corromper o burlar nuestro sitio.

Penetración interna. Es aquella que llevan a cabo usuarios autorizados del sistemas de computadoras que no están autorizados al acceso a los datos que se están comprometiendo, está es la intrusión más común[6] y de esto se hablará en el siguiente capítulo.

Abuso de recursos. Se define como el abuso del manejo de información que un usuario lleva a cabo sobre los datos o recursos de un sistema al que está autorizado su acceso. El abuso puede ser modificación, alteración o revisión del sistema. El conjunto de actividades anómalas se cataloga de la siguiente manera:

Intrusivas pero no anómalas. Se les denomina falsos negativos o errores de tipo I. En este caso la actividad es intrusiva pero como no es anómala no conseguimos detectarla. Se denominan falsos negativos porque el sistema erróneamente indica ausencia de intrusión.

Actividades no intrusivas pero anómalas. Se denominan falsos positivos o errores de tipo II. En este caso la actividad no es intrusiva, pero como es anómala el sistema decide que es intrusiva. Se denominan falsos positivos, porque el sistema erróneamente indica la existencia de intrusión. Estas actividades no son deseables, porque dan una falsa sensación de seguridad del sistema, el intruso en este caso puede operar libremente en él.

Ni intrusiva ni anómala. Son negativos verdaderos, la actividad es no intrusiva y se indica como tal. Los falsos positivos se deben minimizar, en caso contrario lo que puede pasar es que se ignoren los avisos del sistema de seguridad, incluso cuando sean acertados.

Intrusiva y anómala. Se denominan positivos verdaderos cuando la actividad es intrusiva y se detecta. Los detectores de intrusiones anómalas requieren un elevado gasto computacional, porque normalmente se siguen varias métricas para determinar cuánto se aleja el usuario de lo que se considera un comportamiento normal.

Uno de tantos males son los *bugs*, donde el administrador debe asumir que existen y debera convivir con ellos y de preferencia reparar, por ejemplo. Un sistema operativo que no es óptimo en cuestión de seguridad, es irix de silicon graphics, pero muchas

[6]En primer instancia se deja implícito qué significa estar adento y afuera de nuestro sitio.

aplicaciónes solo son creadas para dicha sistema, asi que se puede minimizar riesgos al introducirlo en un *firewall* [101] o invirtiendo en sistemas de seguridad más robustos.

1.10. Sistema Detector de Intrusos (IDS)

Se pueden definir como los sistemas de detección de intrusos [102], ya que analiza los servicios y agentes tanto ajenos como residentes estos pueden estar implicados en un acto malicioso, así que para deslindar responsabilidades el administrador (en caso de que la administración sea centralizada) o los encargados, utilizan una serie de programas y utilerías para una correcta auditoría del sistema además de verificar los 4 puntos esenciales de la seguridad ya mencionados.

La detección de intrusos significa manejar herramientas inteligentes y automáticas para detectar intentos de intrusión en tiempo real. Dichas herramientas se llaman Sistemas de Detección de Intrusos y existen dos pautas básicos de sistemas de detección de intrusos. Está basado en reglas almacenadas en bibliotecas y bases de datos de ataques y firmas responsables de ataques conocidos.

Funciona un IDS cuando se encuentra con un criterio o norma que se etiqueta como un intento de intrusión esto en una pequeña base de datos o reglas. La desventaja es que su base de datos y mantenimiento deben de ser actualizadas constantemente. Si las reglas son muy precisas y el ataque viene de forma modificada el ataque se produce. En está metodología existen dos técnicas: prevención y reacción.

La técnica preventiva, es la escucha de trafico en la red, donde si se oye una actividad de paquetes comparada con su base de datos y si resulta un ataque, actúa de manera que no afecte al sistema, esto significa que en el método preventivo el testigo principal es nuestro detector de intrusos. Por otra parte el método de reacción, inspecciona los *logs* o bitácoras, donde encuentra una actividad sospechosa, el sistema actúa de manera que se fracture el ataque.

La otra pauta [110] es empleando técnicas de reconocimiento de firmas y a esto se le integra inteligencia artificial. Su principal desventaja es su elevado costo y por el momento solo están en el entorno de investigación. Son difíciles de conservar y requieren conocimientos avanzados de matemáticas y estadística.

El IDS trabaja tanto en el servidor como en la red. Los hay tanto libres como comerciales y a su vez están divididos de software exclusivo como de hardware, no es necesario pero un

funcionamiento es optimo con *firewalls* y antivirus.

El software de detección de intrusos (IDS) es otro ingrediente esencial en el ambiente de seguridad del Web. Mientras actúe como una barrera protectora y monitorea contra intentos de intrusión. está técnica es relativamente nueva y su tecnología y diseño aún continúa en desarrollo. Un servidor con un IDS estratégicamente colocado, se transforma en un factor crítico que proporciona protección adecuada para un sitio seguro.

Los sistemas IDS se diseñan para complementar las capacidades del *Firewall.* Extienden sus capacidades al manejo del monitoreo de la actividad de la red, examinando los paquetes de mensajes para los casos de actividad anormal, identificando patrones de ataques conocidos y su mal uso. También proporciona mecanismos de alerta para atraer la atención de manera oportuna. De la misma forma que el *Firewall,* los sistemas IDS deben colocarse en forma estratégica en la red para aumentar al máximo su efectividad.

IDS [104] usa un algoritmo que puede discriminar de forma precisa entre usuarios (basándose en su comportamiento), o de la red (a través de revisiones de ataques ya realizados), mediante reglas que permiten generar bitacoras y/o llamada de atención en tiempo real al administrador; estas actuan ya sea como escuchas en la red o como sentinelas[7] localmente en la revisión de bitacoras de los usuarios o en las actividades normales del sistema.

1.11. Bitácoras del Sistema

Son archivos generados por el sistema, en donde se almacena todos los errores y sucesos importantes en donde los programas y servicios (exceptuando una pequeña parte de creación de terceros) colocan una bitácora de actividades que se realizarón, describiéndonos al usuario, lo que realizó, desde donde se conectó, día, hora y hasta duración de la conexión. Esto se ejecuta por el proceso de *syslog.* Es importante seguir una adecuada configuración para el control de los *logs* que generan un sistema. Muchas alertas se generan con excesiva frecuencia y acaban ocupando gran cantidad de espacio en disco duro. Esto nos puede ocasionar problemas como caídas del sistema o de servicios por una mala planeación de instalación. Existen múltiples herramientas que pueden filtrar la gran cantidad de actos realizados en el sistema. Es uno de los varios métodos de detección porque establece un *modus operandi* de las actividades y pueden llevar a la detección. Al utilizar herramientas de auditoría, los administradores tiene, instrumentos esenciales para detectar intrusos.

[7] sentinela: programa activo como el ya mencionado `syslog` —

El proceso *syslog* consta de:

- *syslogd* (el demonio, que se configura en `/etc/syslod.conf.`).
- Las funciones de las bibliotecas `openlog, syslog, closelog.`

Logger es un programa que permite generar log[8] a los usuarios desde la línea de comandos. Las ubicaciones de las bitacoras pueden variar dependiendo del tipo de Unix, generalizando se encuentran en /usr/adm por las primeras versiones de Unix. /var/adm por versiones de Solaris, Linux y BSD. /var/log usado por algunas versiones de Solaris, Linux y BSD. Por ejemplo la ubicación de las bitacoras del kernel en sus distintos tipos de archivos.

acct (BSD)/pacct(ATT): Guarda las instrucciones ejecutadas por cada usuario. Es actualizado por el kernel.

wtmp: Provee un registro permanente de las conexiones y desconecciones que los usuarios establecen. También guardan las veces que se apaga (shutdown) y arranca (startup) el sistema. En algunos sistemas se guarda en el directorio `/etc`. **fd2log:** Guarda los errores producidos por el sistema de acontecimientos en un sistema ATT.

lastlog: Guarda los datos de la última conexión de los usuarios, y en algunos sistemas también guarda las fallas.

utmp: Guarda un registro para cada usuario conectado al sistema.

messages: Guarda los mensajes enviados a la consola y en algunos sistemas todo lo generado por el programa syslog.

shutdownlog: Registra las razones por las cuales se ha realizado un shutdown. Es actualizado por la instrucción shutdown.

sulog: Guarda todas las instruciones de la instrucción su. El acceso del root vía la utilería su.

authlog: Guarda los autorizaciones. Es actualizado por las instrucciones su, password, login y shutdown.

timed.log: Guarda un registro de la instrucción settime.

[8]log: es el nombre que se le asigna en inglés a las bitácoras del sistema.

sudo.log: Guarda los accesos del root vía la utilidad su del programa de dominio público sudo[9]. La ubicación y el nombre de qué acción se realizó en los registros depende de cómo se va a configurar en la instalación; por omisión lo guarda en la bitacora de security.

Registros relacionados en el sistema de impresión

lpacct: Guarda los llamadas al sistema de impresión en BSD. Es actualizado por el daemon[10] lpd.

lpd-errs: Guarda los errores del sistema de impresión BSD. Es actualizado por el daemon lpd.

Registro relacionados al módem

aculog: Guarda las peticiones al uso del módem[11]. Es actualizado por los programas tip y uucp.

Registro relacionados con correo electrónico

mqueue/syslog: Guarda un registro del correo electrónico. Es actualizado por el programa sendmail.

mqueue/POPlog: Guarda un registro de las conexiones realizadas con el protocol POP. Actualizado por el programa popper.

Registros relacionados con connexiones remotas

uucp/LOGFILE: Registra los contactos via UUCP en un entorno BSD. Es actualizado por la instrucción uucico.

uucp/SYSLOG: Registra las transferencias vía UUCP en un entorno BSD. Es actualizado por la instrucción uucico.

ftp.log/xferlog: Guarda las conexiones realizadas vía ftp.

[9]Sudo: es una utilería que puede servir para descentralizar la administración por ejemplo no es necesario dar el password o clave de acceso de root al administrador de base de datos o el webmaster para reiniciar el servicio.

[10]daemon: Así se le denomina al programa que brinda un servicio en Unix, existen otros como el de mail.

[11]Modem: dispositivo de conexión a redes a través de líneas telefónicas, no es la única manera de conectarse.

tcp.log: Guarda un registro de todas las conexiones tcp. Es actualizado por la instrucción tcpd.

gated.log: Guarda las rutas de red. Es actualizado por el daemon gated.

news/news: Guarda las transacciones realizadas por las news. Es actualizado por las instrucciónes innd.

news/*log: Guarda la actividad del lector de news. Es actualizado por la instrucción nnrpd.

Registro del servidor de X Windows[12]

X0msgs: Registro del servidor X. Actualizado por el programa X11.

xdm-errors: Guarda los errores del administrador del X display. Es actualizado por el xdm.

Registro de los servicios más usuales

httpd/*_ log: Guarda la actividad del servidor de Web. Es actualizado por el daemon httpd.

majordomo.log: Guarda los registros de las listas de distribución controladas por el programa de dominio público majordomo. La ubicación y el nombre del responsable del registro depende de cómo se va a configura en la instalación.

Los fabricantes (SUN, HP, Silicon Graphics, AIX) colocan los archivos de registro en directorios distintos del sistema y los asocia con otros nombres. Por lo tanto, se deberan hacer hincapié en los archivos de arranque del sistema (`/etc/rc`[13], /etc/init.d o xientd[14]) y el archivo de configuración del syslog (`/etc/syslog.conf`).

Si deseas mayor información de los archivos de registro por los sistemas operativos Solaris 2.x, HP-UX 10.x, IRIX 6.x y AIX 3.x, se les recomienda revisar el Anexo bitacoras o la documentación de tu respectivo Sistema Operativo.

[12]servicio de entorno de ventanas

[13]/etc/rc: en /etc se encuentran los archivos de configuración y rc es un directorio donde se ubican los scripts de arranque del sistema

[14]independientemente de texttt/etc/rc/se encuentra un daemon encargado de la administración de servicios

Capítulo 2

Atacantes

En el capitulo, expongo una serie de enfoques de los protagonistas, expongo mi hipotesis de quienes son los atacantes o programas que irrumpen en los sistema de cómputo. En el caso particular de atacantes sus grupos, su preparación, de donde obtiene el conocimiento, técnicas y uso de herramientas. Todo esto hasta llegar a su objetivo, el control de un sistema de cómputo ajeno a su propiedad.

Debemos recordar que el concepto de Seguridad es relativo, no existe una prueba total contra engaños, sin embargo existen niveles de seguridad mínimos exigibles. Este nivel dependerá de un análisis de los riesgos que estamos dispuestos a aceptar, sus costos y de las medidas a tomar en cada caso. Para ubicarnos en la vida real, veamos los datos obtenidos en marzo de 2003 por la consultora Ernst & Young 8 sobre 273 empresas de distintos sectores de actividad y países.

- El 40 % de las empresas estudiadas consideran como un problema grave la seguridad informática.
- El "gasto" en Seguridad Informática oscila entre el 4 % y el 10 % del gasto total informático.
- El 83 % de las empresas reconoce no haber emprendido nunca acciones legales después de un ataque.
- El 72 % se muestra reacia a admitir que sus sistemas han sido saboteados.
- El 79 % cree que existen mayores probabilidades de sufrir un ataque informático procedente del exterior.

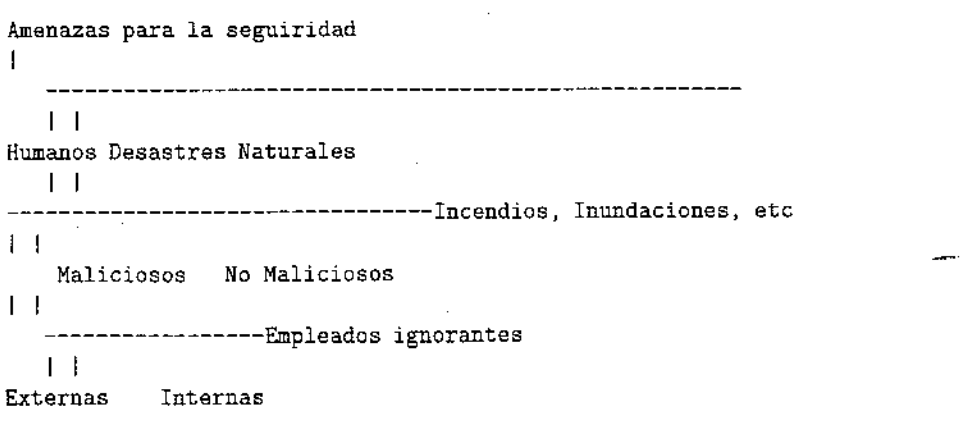
- El 66 % consideran a la Seguridad y Privacidad de la información el impedimento principal para el crecimiento del comercio electrónico.
- El 80 % manifestó no haber experimentado un ataque por intrusión durante el año anterior; pero sólo el 33 % indicó su capacidad para la detección de dichos ataques.
- Sólo el 39 % hace uso de software estándar de seguridad y el 20 % de este total hace uso avanzado de estas herramientas.

Ante el riesgo existente un administrador de servicios de red solo se puede:

- Minimizar la posibilidad de su ocurrencia.
- Reduciendo al mínimo el perjuicio producido, si no ha podido evitarse que ocurriera.
- Diseñar métodos para la más rápida recuperación de los daños experimentados.
- Corrección de las medidas de seguridad en función de la experiencia recogida.

2.1. Amenazas

Cabe definir amenaza [120], en el entorno informático, como cualquier elemento que comprometa al sistema.



Se considera amenaza, cualquier ejecución que no esté estrictamente permitida, la cual pueda ser maliciosa al sistema, al sitio o que irrumpa la seguridad del servidor. Puede ser considerado ataque cualquier modificación no permitida, dentro de las politicas de seguridad del sitio; puede esto desencadenar la ejecución a través de actos(ingenieria social o ejecución de programas) no legales obteniendo permisos de super usuario o administración, puede sustituir y/o manipular recursos del sistema para fines no permitidos o no establecidos por la administración.

Generalmente, los atacantes manipulan errores de sistemas, que están constituidos por ejemplo con configuraciones con condiciones iniciales(por default), por ejemplo de CGI's[1] que no fueron removidas por el administrador, esto considerado como descuidos. Otros ejemplos son dispositivo mal configurados, servicio de red o uso del equipo de producción en un lugar crítico en el sitio.

Otro grupo de atacantes que no pertenecen al grupo de trabajo del servidor, son los que quieren utilizar el sistema o los recursos como usuarios legítimos; utilizar habilidades especiales como la manipulación social o ingeniería social, compartir claves o malas políticas del sitio para la asignación de *passwords*. Esto sólo influye si el posible atacante se encuentra en nuestro lugar de trabajo o quiera asociarse a él.

En los casos en que un usuario tenga excesos de permisos, puede ser conflictivo a la administración y al sistema, suponiendo que la secretaria desee ingresar a la base de datos para hacer una modificación de los mismos también que usuarios externos puedan ingresar a la cuenta de la secretaria y alterar los correos del jefe, por ejemplo. Las ventajas son enormes, pero los inconvenientes pueden ser diversos.

El análisis de un sistema [87], no solo se limita al administrador o al oficial de seguridad, también grupos de individuos en internet prestan gran atención a estos detalles y no se centra en un sistema de cómputo específico como Unix sino va ir dependiendo de qué los motiva y sus metas. Las personas, tienen caracteristicas distintas, la ventaja de estos sujetos es que para hacer un robo de dinero no es necesario cargar con un arma y presentarse en el banco, sino con una simple transacción comercial. Lo que pretenden la gran mayoria de los intrusos es tener la cuenta del superusuario. Las desventajas son enormes por lo que no solo debemos de protegernos fisicamente, tambien a nivel logico.

[1]CGI: sección de srvidor de web, donde uno puede ejecutar aplicaciónes desde sitios remotos

2.2. Caracterísicas de la vulnerabilidad

En cualquier ramo donde esté inmersa la computación es necesario el análisis [123], la garantía de la seguridad de la información y la necesidad de un buen admistrador para coordinar los servicios de cómputo. Si los ejecutivos de un negocio o jefes de áreas no saben a quién otorgarle permisos se crea un peligro o amenaza latente contra la información.

Definiendo, amenaza, a la condición del entorno del sistema de información que, dada una oportunidad, podría dar lugar a que se produjese una violación de la seguridad (confidencialidad, integridad, disponibilidad o uso legítimo de recursos). La política de seguridad y el análisis de riesgos habrán identificado las amenazas que han de ser contrarrestadas. Dependiendo del diseñador del sistema de seguridad se debe especificar los servicios y mecanismos de seguridad necesarios. Las amenazas a la seguridad en una red pueden caracterizarse modelando el sistema como un flujo de información desde una fuente, como por ejemplo un archivo o una región de la memoria principal, a un destino, por ejemplo otro archivo o un usuario.

Un ataque está catalogado dentro de las siguientes cuatro categorías generales o ataques son las siguientes:

Interrupción: un recurso del sistema es destruido o se torna no disponible. Este es un ataque contra la disponibilidad. Ejemplos de este ataque puede ser la destrucción de un elemento como hardware, un disco duro, cortar una línea de comunicación o deshabilitar el sistema de gestión de archivos.

Intercepción: una entidad no autorizada consigue acceso a un recurso. Este es un ataque contra la confidencialidad. La entidad no autorizada podría ser una persona, un programa o un computadora. Ejemplos de este ataque puede ser ejecutar una línea para obtener datos que circulan por la red y la copia ilícita de archivos o programas (intercepción de datos), o bien la lectura de los encabezados de paquetes para engañar la identidad de uno o más de los usuarios implicados en la comunicación observada ilegalmente (intercepción de identidad).

Modificación: una entidad no autorizada no sólo consigue tener acceso a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad. Ejemplos de este ataque son el cambio de valores en un archivo de datos, alterar un programa para que funcione de forma diferente y modificar el contenido de mensajes que se copia o mueve a través de la red.

Fabricación: una entidad no autorizada inserta objetos falsificados en el sistema. Es un ataque contra la autenticidad. Ejemplos de este tipo de ataque son la inser[illegible]

mensajes espurios en una red o añadir registros a un archivo.

Y al clasificar las formas de ataque, se consideran dos métodos:

Ataques Pasivos: El atacante no altera la comunicación, sino que únicamente la "escucha" o monitoriza, para obtener información que está siendo transmitida. Sus objetivos son la intercepción de datos y el análisis de tráfico. Generalmente se emplean para:

- Obtención del origen y destinatario de la comunicación, a través de la lectura de las cabeceras de los paquetes monitorizados.
- Control del volumen de tráfico intercambiado entre las entidades monitorizadas, obteniendo así información acerca de las actividades o inactividades inusuales.
- Control de las horas habituales de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los periodos de actividad.

Es posible evitar el éxito, del atacante mediante el cifrado de la información y otros mecanismos que se verán posteriormente. Todo ésto puede otorgar información importante sobre el sistema, así como permitir la aproximación de los datos que contiene. En los ataques pasivos el atacante no altera la comunicación, sino que únicamente el monitoreo, para obtener información que está siendo transmitida. Su objetivo es la interceptar datos y análizar de tráfico, una técnica más sutil para obtener información de la comunicación[2].

Ataques Activos: Estos ataques implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos. Generalmente son realizados por hackers, piratas informáticos o intrusos remunerados y se los puede subdividir en 5 categorias

- Interrupción
- Intercepción
- Modificación
- Fabricación
- Destrucción: es una modificación que inutiliza a l objeto en cuestión.[3]

[2]En este rubro puede ingresar termino *backdoor* o puerta trasera el cual se va activar cuando el atacante quiera y durante su estancia no sera persivido por el administrador o usuarios

[3]Fuente HOWARD, Jhon D. Thesis: An Analysis of security–on the Internet 1985-1995

servicio.

Un caso de negación de servicio fue lo que sucedió en febrero del 2000 Amazon, donde la intrusión con programas *roolkit* mantuvo a la empresa durante 2 días fuera de comunicación. Otro de tantos métodos es escuchar la red, o mejor conocido como implantar un *sniffer*.

2.3. Atacante

Se le llama atacante o intruso a la persona que accede (o intenta acceder) sin autorización a un sistema ajeno, ya sea en forma interna del sitio o servidor o externo al sitio. Los intrusos podríamos clasificar desde el punto de vista de conocimiento:

1. El 80 % "nuevos intrusos" los cuales bajan programas de Internet y prueban programas para introducirse a los sistemas de computo.
2. El 12 % "son más peligrosos", saben compilar programas aunque no saben programar. Prueban programas, conocen como detectar que sistema operativo está usando la víctima prueban las vulnerabilidades del mismo e ingresan por ellas.
3. El 5 % es gente con amplios conocimientos y define sus objetivos.
4. El 3 % restante entra a determinados sistemas sólo por la información que necesitan.

En promedio general, las personas que están en la última categoria tuvieron que pasar al menos 4 años, de conocimiento de software, programas y análisis ed sistema.

2.4. Tiempos en que uno puede ser atacado

Las amenazas puede ser analizadas en tres momentos: antes del ataque, durante y después del mismo. Estos mecanismos conformarán políticas que garantizán la seguridad del nuestro sistema informático.

- Prevención(antes): mecanismo que aumenta la seguridad (o fiabilidad) de un sistema durante su funcionamiento normal. Por ejemplo el cifrado de información para su posterior transmisión.

- Detección (durante): mecanismo orientados a revelar violaciones a la seguridad. Generalmente son programas de auditoria o monitores de proceso.

- Recuperación (después): mecanismos que se aplican, cuando la violación al sistema ya se ha detectado, para retornar éste a su funcionamiento normal. Por ejemplo recuperación desde las copias de seguridad (backup) realizadas.

2.5. Motivación de los atacantes

Revisando el entorno, tomemos en cuenta que los sistemas de cómputo se distribuyenentre hogares, empresas, instituciones, etc. No son iguales. Los usuarios y sus costumbres computacionales no son las misma(ejemplificando, no todo compran en internet), los sistemas operativos no son los mismos, ni siquiera el sistema seguridad optimo o actualizado.

Entonces, que motiva a violar los sistemas de cómputo, que orilla a llegar a hacer actos corruptos o vandálicos a usuarios, instituciones, etc. Hasta donde se podria considerar un ataque, o que hayamos sido presas de uno.

Diversos autores dedicados al tema de la psicologia y seguridad computacional clasifican a los atacantes en categorías, estas divisiones son hechas por los motivos de intrusión y por las habilidades o uso de herramientas.

Se dice que los atacantes son personas ociosas que disponen de mas tiempo y de mejor equipo de cómputo que el nuestro, por ende, tienen procesos más ágiles lo que puede estar en nuestra contra por los servicios que prestamos en Internet en nuestra red local. Los motivos por los que los intrusos nos afectan son los siguientes.

2.5.1. Ego

Existen comunidades bien organizadas que son considerados como el nuevo terrorismo. Estas comunidades no tienen que matar un número alto de personas para llamar la atención, en cambio el terrorismo cibernético se aprovecha de la mala estructura administrativa y computacional de múltiples lugares en Internet.

Lo que tratan de hacer estos atacantes es llamar la atencion de la sociedad de un problema o una necesidad de ser escuchado. por eso burlan los sistemas de cómputo. En ocasiones

las élites de atacantes invitan a otros usuarios novatos que han ingresado a sitios con una administración y seguridad alta. Buscan alardear por haber ingresado a un sitio y que el intruso tiene el mando en ese lugar, o si realizó este acto delictivo, mencionar que "yo que instalé esto en `victima.hackeada.com`, imagínense que puedo hacer con sus bases de datos".

Otra manera de satisfacer su ego es al hacerse publicidad a través de páginas web. La lista de ataques exitosos es larga, la mayoría de los novatos trata de sobresalir e impactar en este servicio. Para ejemplificar sería útil revisar la siguiente pagina en: `http://www.atrition.com` donde nos dan listas muy claras de los atacantes y su manipulación visual, donde dejan la firma de los involucrados.

2.5.2. Medio de protesta

Muchos de los atacantes se descubren por su modo de operar. Realizan protestas socio culturales o politicas y tratan de tomar como foro de expresión por ejemplo un *CHAT* o una página *web* de una institución pública o privada. Por ejemplo el caso de Casa Blanca, sin olvidar los pequeños ataques de usuarios locales que están en contra de un maestro o un administrador o contra el jefe directo o corporativo, donde prueban sus conocimientos, para desmembrar la moral de la víctima.

2.5.3. Curiosidad o diversión

Son aquellos que realizan lectura de servicios a bloques grandes de direcciónes IP's, como en las universidades o sitios gubernamentales o de investigación, que consideran más apropiados porque en estos lugares se encuentran computadoras más poderosas y la red es más rapida comparándola con su modem o ISDN.

¿De donde provienen tanto curioso? De todo el mundo, del Reino Unido, China, Pakistán, Argentina entre otros. Buscan servidores sin administración y/o con vulnerabilidades recién explotadas.

Quienes están más especializados ocupan técnicas de *Worm*(gusanos) donde la aplicación realiza una exhaustiva búsqueda para ingresar a través de *login* y haciendo ataques de fuerza bruta contra los password. Esto aprovechando que una gran cantidad de administradores crean logins con nombres tan comunes como webmaster, Pérez González o nombres conjugados como jrodriguez tsmith. o en base a los servicios que prestan como acceso web o de

impresión.

2.5.4. Venganza

La venganza contra una empresa u organización, ya sea que tenga contacto directo o sea el atacante manipulado hasta personalizar esa ira sin tener relación directa, por compañerismo. Como por ejemplo por odio contra una empresa que está monopolizando un servicio como el teléfono o la luz, contra un gobierno, etc.

También por venganza contra el administrador por el uso restringido de las cuentas, o contra la organización o institución donde está el conjunto de servicios de Internet, o porque envidias.

2.5.5. Para obtener beneficios económicos

Es el problema más peligroso de cualquier negocio o empresa. Pensemos en un comercio electrónico. donde deberían tener SSL(Secure Soket Layer): Mecanismo de seguridad donde el clienete y el servidor generan llaves RSA para la autenticación del origen y destino. En su portal web de comercio, para que puedan transferir movimientos financieros, pero existe un *sniffer* instalado en su red, y así se obtienen los datos de trasferencia. Lo menos dañino es cancelar el producto (esto no ocurre), pero sí el cargo a la tarjeta de crédito de la víctima y una posible adjudicación de otros bienes y servicios.

Un caso ecónomico más fuerte es el desvío de fondos, o la anulación de un adeudo al banco o hacia un particular; la modificación de permisos para ejercer otra aplicación no permitida; el incremento de salario en una pequeña empresa. Como estos ejemplo pueden existir más.

Motivación

Ocurre cuando un intruso desea modificar, en forma parcial o total,datos ya sea en una base de datos o en un archivo, por ejemplo un ingreso a un banco y pueda modificar los datos, o en el momento de la transación entre 2 personas o al modificar archivos del sistemas basicos como un acceso remoto al servidor (`rhost, shost,`) etc. Hasta alteraciones totales y crear confiicos entre los usuarios.

2.5.6. Utilización de recursos

Es cuando se emplean los recursos de procesamiento de la máquina atacada para ejecutar aplicaciones, para afectar otros sitios. Por ejemplo el gusano "ramen" que ejecutaba aplicaciones con fines programador. Casos comunes son los ataques a servidores o PC's, cuando el equipo victima.hackeada.com es de mejor o mayor procesamiento que el atacante, esto se realiza con fin de ingresar a ese equipo para ocupar procesos que el usuario no utiliza, y ser canalizado por un ataque mayor. Este proceso no sólo se limita al tipo de máquina o procesador sino también al disco duro y sus servicios que realiza por la red. Un caso tipico es el de los `ftp://piratas.de.software.com` donde alojan software, juegos, entre otras cosas en sitios de ftp públicos donde se corrompe la licencia.

Lectura y ejecución de archivos

El enemigo no siempre está en el exterior, el mayor número de ataques están constituidos por socios de nuestros servicios locales o dentro de nuestra organización. Las políticas de seguridad deben de poner en práctica barreras internas porque cualquier fractura que logre afectar la confidencialidad de archivos puede implicar que los atacantes no solo obtengan información del sistema sino un robo de datos o lectura de secciones de bases de datos no permitidas para un tipo de departamento.

2.6. Efectos de la vulnerabilidad

Las vulnerabilidades resultante en un daño menor pueden ser reparables, por ejemplo en caso de un sistema operativo, un servicio o aplicación, pero lo principal son los clientes, debido a que se puede perder la información y/o la confianza y va a depender del daño adquirido.

2.6.1. Pérdida de confianza

Son los casos particulares donde el atacante logra sustituir a un individuo legal en una aplicación o servicio, para ello pudo haber burlado la seguridad o adquirir de manera ilícita el acceso. A su vez logra que el administrador pueda inculpar el acto cometido hacia otro individuo. Otro ejemplo es cuando un intruso logra ingresar al servidor "A" y desde ese

punto coloca un sniffer y un *scanner* de puertos que puede afectar a un conjunto de IP's, el servidor "B". En este ataque implica a " A" como responsable.

2.6.2. Caídas del sistema y de los servicios

Por otra parte, toda caída del sistema o servicios debe de ser revisada, encontrar el motivo de la causa y su origen, no se deberá menospreciar el origen de esa inesperada actitud del sistema; en ataques a gran escala, los atacantes están utilizando spoofing[4], ruptura de servidores DNS, Spam para servicio de mail.

Existen dos tipos de caidas del sistema:

DOS Denegación de servicio : Es crear un mal funcionamiento de un servicio en sus capacidades en forma excedida del sistema.

DDOS Denegación de servicio distibuido Es la caída del sistema por aplicaciones externas a el sistema, realizando a la vez una múltiple petición a un solo servicio.

En un problema de órden económico, ?cuánto perdería una empresa por ejemplo Amazon si se desconectara durante 2 días?, o si `http://www.yahoo.com/` negara servicios, porque sus servidores no soporta el nivel de carga de peticiones, debido a que está enfrentando un ataque de excesos de clientes que saturó el servicio de cientos de máquinas.

Estos problemas dañan la imagen de la empresa, si uno se coloca del lado del consumidor y no puede ingresar a un sitio donde desea comprar un producto, en la mayoria de los casos lo que hace es buscar otra opción de compra, y esto expresado en dinero representa una pérdida para la compañia.

2.7. Riesgos o efectos

La seguridad de la información y por consiguiente de los equipos informáticos, es una cuestión que llega a afectar, incluso, la vida privada de los seres humanos; de ahí que resulte obvio el interés creciente sobre este aspecto de la nueva sociedad informática.

[4]Spoofig: enmascaramiento de dirección IP, simular en la red recuursos no permitidos

Ladrones, manipuladores, saboteadores, espías, etc. reconocen que el centro de cómputo de una institución es su sistema nervioso central, y normalmente tiene información confidencial que, a menudo, puede ser vulnerable a algún ataque.

Mantener a buen resguardo la información tiene dos aspectos.

El primero consiste en negar el acceso a los datos a aquellas personas que no tengan derecho a ellos, lo que también se puede llamar protección de la privacidad, si se trata de datos personales y mantenimiento de seguridad en el caso de datos institucionales.

Un segundo aspecto de la protección es garantizar el acceso a todos los datos importantes a las personas que ejercen adecuadamente su privilegio de acceso, las que tienen la responsabilidad de proteger los datos que se les ha confiado.

En general, la protección de los datos requiere ejercer un estricto control sobre la lectura, escritura y empleo de esa información. Para ser más eficientes en la protección de los datos se debe tener siempre presente, el mantenimiento de la privacidad y la seguridad del secreto. El secreto se logra cuando no se tiene acceso a todos los datos sin autorización, por ejemlplo, si deseamos crear *oscurantismo* como base de nueestra seguridad, necesitaremos negar mucha informacion, por ejemplo uname, es un comando que nos da la opcion de saber el sistema operativo, plataforma(ejemplo: windows, i386 linux, sparc, power pc), version de kernel, sera tan necesario entregar este comando a los huespedes o a internet. Una adecuada privacidad puede lograrse cuando los datos que puedan obtenerse no pueden enlazarse a individuos específicos o no pueden utilizarse para imputar hechos acerca de ellos.

Por otra parte, es importante incorporar dispositivos de seguridad cuando se realice el diseño del sistema en vez de añadirlas después. Los diseñadores de sistemas deben entender que las medidas de seguridad han llegado a ser criterios de diseño tan importantes como otras posibilidades funcionales y se deberá pensar en agregar funciones en el incremento de los costos, después de desarrollado un Sistema de Información.

Los equipos de cómputo son posesiones muy valiosas de las empresas y están expuestos al "robo", de la misma forma que lo están las piezas de inventario e incluso el dinero. Es frecuente que los operadores utilicen la computadora de la empresa para realizar trabajos privados para otras organizaciones y, de esta manera, robar tiempo de máquina. La información importante o confidencial puede copiarse fácilmente. Muchas empresas invierten millones de dólares en programas y archivos de información, a los que dan menor protección que la que otorgan a una máquina de escribir o una calculadora. El software, es una propiedad que puede ser sustraída fácilmente, en discos riguidos o extraibles sin dejar ningún rastro.

2.7.1. Fraude

Cada año, millones de dólares son sustraídos de empresas y, en muchas ocasiones, las computadoras han sido utilizadas para tal propósito. En realidad, el potencial de pérdida a través de fraudes y los problemas de prevención y detección del fraude, van en aumento en sistemas computacionales. Sin embargo, debido a que ninguna de las partes implicadas (compañía, empleados, fabricantes, auditores, etc.), tienen algo que ganar, sino que más bien pierden en imagen, no se da ninguna publicidad a este tipo de situaciones.

Las tres principales áreas donde se produce el fraude son:

1. Manipulación de información de entrada, fácil de realizar y muy difícil de detectar, ya que los métodos de validación de entrada son simples y en general, conocidos por un gran número de personas de la empresa.
2. Alteración o creación de archivos de información. Se alteran los datos directamente del archivo o se modifica algún programa para que realice la operación deseada.
3. Transmisión ilegal. Interceptar o transferir información de teleproceso.

Factores que conducen al fraude con computadoras

- Baja moral entre el personal. Los empleados en los departamentos de procesamiento de datos y de usuarios de la computadora, muestran falta de disciplina respecto a las precauciones de seguridad y en mantener una operación ordenada y sistemáticamente realizada.
- Documentación deficiente. La documentación del sistema está incompleta, anticuada y desordenada. Sólo el diseñador del sistema tiene una idea verdadera de lo que hace el sistema.
- Personal innecesariamente atareado todo el tiempo. Empleados con pocos permisos para ausentarse, en la misma función, durante largo tiempo y rara vez toma vacaciones (Una vez que un fraude está en marcha, el delincuente se mantiene en continúa vigilancia para evitar ser descubierto).
- Deficiente administración de la operación. Falta de control de documentos y de procedimientos de autorización, regulando cambios del sistema y alteraciones a los archivos de datos. Falta general de control del sistema.

- Alta incidencia de equivocaciones de la computadora. Errores creados por un diseño deficiente del sistema hacen que el personal y gerentes acepten errores susceptibles de "inculpar a la computadora".

2.7.2. El Sabotaje

El peligro más temido por los centros de Procesamiento de Datos, es el sabotaje. Empresas que han intentado implementar programas de seguridad de alto nivel, han encontrado que la protección contra el saboteador es uno de los retos más duros. El saboteador puede ser un empleado o un sujeto ajeno a la propia empresa.

Los imanes son herramientas muy empleadas, aunque las cintas estén almacenadas en el interior de su funda de protección, una ligera pasada y la información desaparecerá. Una habitación llena de cintas puede ser destruida en pocos minutos. Los Centros de Procesamiento de Datos pueden ser destruidos sin entrar en ellos. Suciedad, partículas de metal o gasolina pueden ser introducidos por los conductos de aire acondicionado. Las líneas de comunicaciones y eléctricas pueden ser cortadas, etc.

2.8. Clasificación de atacantes

A comienzos del 2004, más de 40.000 sitios de Internet fueron comprometidos por algún tipo de ataque y alrededor del 10% de los usuarios conectados a Internet (aprox. 3.000.000 de personas), han realizado algún acto de intromisión a computadoras conectadas a la red.

Los atacantes comparten ciertas características como por ejemplo, querer ocultarse y no ser atrapados, y si logran su meta de intrusión desean conservarla, así que colocan puertas traseras para reingresos posteriores y/ o utilizan el sistema para poder ingresar a otros, obteniendo con ello permanecer en el anonimato. La mayoría alardea de sus intrusiones con sus colegas así como su forma de ingreso a ciertos sistemas.

Los llamados "hackers [141]" o piratas informáticos ocupan más de mil páginas web para mostrar sus conocimientos a principiantes que quieran adentrarse en el mundo de los virus informáticos. Las páginas muestran en su mayoría diferentes calaveras que van girando, como "links" (enlaces) para entrar en el mundo de cada uno de estos creadores de virus. La veracidad de su información es variada.

Habría muchas formas de clasificar a los habitantes del ciberespacio, pero la más comun es aprovechando los conocimientos de que disponen y del medio en el que circulan. Así, hay usuarios de andar por casa, hay gente más o menos informada sobre el funcionamiento de las cosas, usuarios medios, técnicos, programadores y *hackers*. Definamos estos términos.

2.8.1. Hackers

Los protagonistas de nuestra era tecnológica (referente a los sistemas operativos e Internet) se autodenominan hackers. Se definen a sí mismos como personas que se dedican a programar de manera apasionada y creen que es un deber para ellos compartir la información y elaborar software gratuito. Un hacker es un experto o un entusiasta de cualquier tipo que puede dedicarse o no a la informática. En este sentido, la ética hacker es una nueva moral que desafía la ética protestante del trabajo, tal como la expuso hace casi un siglo Max Weber en su obra clásica. La ética protestante y el espíritu del capitalismo, y que está fundada en la laboriosidad diligente, la aceptación de la rutina, el valor del dinero y la preocupación por la cuenta de resultados. Frente a la moral presentada por Weber, la ética del trabajo para el hacker se funda en el valor de la creatividad, y consiste en combinar la pasión con la libertad. El dinero deja de ser un valor en sí mismo y el beneficio se cifra en metas como el valor social y el libre acceso, la transparencia y la franqueza.En el libro de Max Weber, nos invita a recorrer las cuestiones fundamentales sobre la vida en la sociedad de la información, a emprender un viaje lleno de sorpresas que nos ayudará a orientar nuestras vidas hacia nuevas y apasionantes perspectivas.

El movimiento hacker es amplio e importante. Es controvertido y polémico. Como cualquier aspecto de la vida presenta aspectos positivos y otros criticables. Su lucha por la libertad, por la superación de las restricciones, por la no sujeción a las normas, sus propuestas individuales son altamente sugerentes. Es inegable el protagonismo histórico del ideario hacker en el nacimiento y desarrollo de los cambios tecnológicos que han dado lugar a las nuevas tecnologías y sus importantes cambios sociales. Sin lugar a dudas Internet, el movimiento a favor del código abierto, singularmente el fenómeno Linux, las propuestas en favor de la libertad de expresión y comunicación, las luchas contra las patentes y leyes restrictivas sin el ideario hacker serían muy distintas y el escenario social que se habría dibujado sin esta mentalidad hacker seria otro.

Indudablemente el movimiento hacker, en su sentido original, tiene un importante estigma social provocado por los crackers que violan sistemas, generan y distribuye virus altamente destructivos, roban información, "derrocan" servidores y realizan todo tipo de actos ilícitos. La mayoría de las gentes, especialmente los medios de comunicación, no saben distinguir

entre los hackers y crackers, englobando ambas actitudes y acciones en el mismo paquete, cuando deberían deslindarse.

El término "hacker" tiende a connotar participación como miembro en la comunidad global definida como "la red". También implica que la persona descrita suele suscribir alguna versión de la ética del hacker.

Desde que se usó por primera vez la palabra hacker, hace más de 10 años, ésta ha sido mal utilizada, mal interpretada y encasillada en un contexto errado. Antes de continuar, aclaremos que el término *hacker* no tiene nada que ver con actividades delictivas, si bien muchos hackers cometen errores, la definición no tiene nada que ver con ello.

En este rubro pueden entrar casi todos los atacantes porque debido a sus orígenes, el intruso es una persona con amplios conocimientos del sistema operativo, sus utilerías y aplicaciones, y a su vez, el conocimiento de sus vulnerabilidades, aunque en grupo no sabe realmente que hace o no tiene planes con el sitio vulnerado. Pero hoy en día, los considerados en este rubro, realizan un trabajo más elaborado, desde un simple escaneo de puertos para saber el modo de ingreso a un sitio protegido, burlando las implementaciones de seguridad con programas que rompan esta protección hasta llegar a su meta, después los programas se distribuyen (con el nombre de *rootkits, rootshell, exploits, worm, etc.*). Tambien, tratan de obtener cuentas bancarias en Bancanet en México o Visa Internacional, o bien reingresar al Pentágono o a empresas corporativas como IBM, Microsoft, Intel, SecureSO, etc.

El mejor libro sobre los verdaderos **hackers** es "hackers", de Steven Levy. Sobre historias de *hackers* y crackers se ha escrito mucho, desde el famoso "El huevo del cucu [161]" hasta otros trabajos de documentación e investigación bastante buenos.

Para una correcta definición necesitaremos conocer el origen de la palabra hacker. [158]

Hack [126] Como verbo, significa tajar, cortar, dividir una cosa en pedazos. También quiere decir alquilarse, venderse o prostituirse.

Como sustantivo: peón, mozo que se alquila, escritor mercenario o, en otra acepción, muesca, corte o tajada. HACKER [Originalmente, alguien que fabrica muebles con un hacha]; hoy en dia se nombra así a:

1. Persona que disfruta con la exploración de los detalles de los sistemas programables y sabe cómo aprovechar sus posibilidades; al contrario que la mayoría de los usuarios, que prefieren aprender sólo lo imprescindible en programación.

2. El que programa de forma entusiasta (incluso obsesiva).

3. Persona capaz de apreciar el "valor del hackeo".

4. Persona que es buena programando de forma rápida. Experto en un programa en particular, o que realiza trabajos usando frecuentemente cierto programa; como en "es un *hacker*de UNIX". (Las definiciones 1 a 5 están correlacionadas, y la gente que encaja en ellas suele congregarse).

5. Experto o entusiasta de cualquier tipo. Se puede ser un "hacker astrónomo", por ejemplo.

6. El que disfruta del reto intelectual de superar o rodear las limitaciones de forma creativa.

7. Malicioso que intenta descubrir información sensible analisando por varios sitios. De ahí vienen "hacker de contraseñas" y "hacker de las redes". El término correcto en estos casos es **cracker.**

Nótese que ninguna definición define al *hacker* como un criminal. En el mejor de los casos, son los incentivadores, probadores y aprobadores de las mejores y más nuevas tecnologías. En el peor, los hackers pueden ser traviesos, perversos y delincuentes curiosos.

Si estendemos como *hacker* al individuo que usa sus habilidades y recursos para invadir sistemas informáticos ajenos, dejamos un hueco en la definición, pues no es tan simple, un *hacker*puede ser un niño travieso, un joven delincuente o un gran profesional contratado por una gran corporación. Lo que sí, no podemos evitar aceptar que están presentes por cientos en el Internet y que en general, están buscando problemas.

Si usted alguna vez soñó con ser un *hacker* y manipular el mundo desde su computadora personal, deje de soñar! ser *hacker*requiere de mucha preparación, poderosos y variados equipos y sobre todo, horas y horas a solas, tecleando códigos ilegibles para otros.

Mas información disponible en:

`http://www.etcetera.com.mx/libro/tres/comp3.htmhttp://www.geocities.com/SiliconValley/B`
`http://www.2600.com/news`

Desde hace tiempo circula por Internet el libro , la "Ética de los hackers", que trata de mantener claras sus motivaciones, objetivos y sobre todo, que trata de mantener alejados a los controladores de la ley.

- El acceso a las computadoras y a cualquier cosa que pueda enseñarte algo acerca de la forma en que funciona el mundo debe ser total e ilimitado.
- Apelar siempre a la imperativa: ¡Manos a la obra!
- Toda información debe ser libre y/o gratuita.
- Hay que desconfiar de la autoridad. Hay que promover la descentralización.
- Los hackers deberán ser juzgados por sus intrusiones, no por falsos criterios como títulos, edad, raza, o posición.
- Las computadoras pueden cambiar la vida para mejorarla.

Existe mucha bibliografía sobre estos "personajes", y aunque no debe servirnos de consuelo, tienen acceso en forma habitual a las redes académicas y comerciales en todo el mundo. Están organizados y realizan congresos anuales donde intercambian experiencias y anécdotas. Entre sus blancos predilectos figuran las empresas telefónicas, los multinacionales, etc. En resumen: un *hacker*es simplemente alguien capaz de manejar con gran habilidad un aparato, no necesariamente una computadora, con el fin de sacarle más partido o divertirse. ¿Qué hay hoy en día que no sea programable? Desde el reloj de pulsera hasta el vídeo o la radio del coche. Y todos esos pequeños aparatos pueden ser programados y "hackeados" para que hagan cosas que se supone que no pueden hacer. La historia no ha reflejado el cambio abismal entre el origen de un *hacker*el cual se olvida de la parte humana para satisfacer la sed de conocimiento o reto que es infundado con conocimientos empíricos, esto me refiero a que sus conocimientos no son cursos de especializacion de redes o de sistemas operativos, pero no se debe de dudar de dichos conocimientos.

2.8.2. Crackers o Vándalos [96]

Son intrusos que realizan actos maliciosos y atacan a servidores por considerar como sus enemigos a los administradores o instituciones que son blanco de sus ataques. Estos atacantes tienen una vida anonima muy corta. Dejan pistas llamativas, y relativamente sencillas de detectar. No gozan de experiencia ni muchos conocimientos.

Cracker: El que rompe la seguridad de un sistema. Acuñado en 1985 por hackers en defensa contra la utilización inapropiada de este termino por periodistas. Falló un intento anterior de establecer "gusano" en este sentido en 1984-1982 en Usenet.

La utilización de ambos neologismos refleja una fuerte repulsión contra el robo y vandalismo perpretado por los círculos de crackers. Aunque se supone que cualquier *hacker* auténtico ha jugado con algún tipo de crackeo y conoce muchas de las técnicas básicas, se supone que cualquier que haya pasado la etapa larval ha desterrado el deseo de hacerlo con excepción de razones prácticas inmediatas (por ejemplo, si es necesario pasar por alto cierto sistema de seguridad para completar algún tipo de trabajo). Hay mucho menos en común entre el mundo de los hackers y de los crackers de lo que el lector mundano, confundido por el periodismo sensacionalista, pueda suponer.

Los crackers tienden a agruparse en grupos pequeños, muy secretos y privados, que tienen poco que ver con la poli-cultura abierta; aunque los crackers a menudo se definen a sí mismos como hackers, la mayor parte de los auténticos hackers los consideran una forma de vida inferior.

En general, los medios han hecho un favor a los hackers al hablar sin conocimientos sobre los asuntos en los que se ven envueltos. Los hackers son muy diferentes de los crackers, son hackers maliciosos cuyo objetivo es introducirse ilegalmente en sistemas, desproteger productos y hacer cosas similares.

Entre las variantes de crackers maliciosos están los que realizan Carding (Tarjeteo, uso ilegal de tarjetas de crédito), *Trashing* (Basureo, obtención de información en botes de basura, tal como números de tarjetas de crédito, contraseñas, directorios o recibos) y *Phreaking o Foning* (uso ilegal de las redes telefónicas). Otras razones por las que se mira con desprecio a los crackers se describen en las entradas sobre cracking y phreaking (crackers telefónicos).

2.8.3. *Score keepers* o Espias

No tienen preferencias por un sitio en especial, su reto es buscar una buena administración, no les importa ni los datos, ni el peso en la sociedad del sitio al que atacan. Toda información obtenida se puede compartir con otros intrusos, en sitios como:

`http://www.hackernews.com`

`http://www.cultdeadcow.com`

`http://www.10pht.com`

`http://www.neworder.box.sk`

Los espias consideran ataque a granel [7], entre mayor sea la dificultad en el ingreso más les interesa llegar a él. Y cuando estan saciados de esa información pueden venderla o intercambiar.

Puede ser toda persona que mediante la intrusión puede perseguir un bien económico como por ejemplo números de tarjetas de crédito, o acceso a ciertas redes. El ciclo de los espías es que si alguien es espía profesional puede ser que otro colega lo esté vigilando a y a su vez otro y así sucesivamente.

2.8.4. *Joy riders o Script kiddies o Lamers*

Un lamer es una persona que no tiene ninguna inquietud por todos estos temas de seguridad sino que lo único que quiere es tener una identificación para entrar a un sistema y formatear el disco duro, o para decirle a un amigo que es un superhacker. Es importante distinguir lamer de newbie o novato. Un novato o newbie es una persona que SÍ que tiene interés en estos temas pero que lógicamente necesita un tiempo de aprendizaje ya que nadie ha nacido aprendiendo.

Estos atacantes son los que más abundan, son los individuos que obtienen algún programa de intrusión o revisión de red (scanner), su finalidad es introducirse a servidores sin importar cuáles, para obtener y/ o modificar datos interesantes, y dejan un rastro de su agresión, esto para difamar sitios o utilizarlos como trampolín hacia otros sitios. Puede ser también un método de ingreso a organizaciones o clanes de " hackers" como páginas modificadas. Aunque no son maliciosos, puede suceder que por error puedan destruir el sistema, solo les atraen sitos conocidos y computadoras poco comunes [116].

2.9. Metodos de ingreso a un servidor

Existen muchas formas de ingresar a los servidores, a continuación describire una lista de las formas más sencillas para vulnerar.

2.9.1. Por un usuario

De algún modo debe obtener la contraseña de un usuario, para esto utilizan desde ataques de ingeniería social[5], el pago a terceras personas para monitorearlo, basado en sus conocimientos sobre sistemas operativos, buscará la manera de poder "actuar" dentro de la red con propiedades de super-usuario.

Metodo de obtiene la contraseña del usuario:

1. Intenta con un logeo de usuario "típicos" con nombre comun("ana", "maria", "jorge", etc) y con contraseñas "elementales"(que no tenga contraseña, el propio nombre del usuario, el apellido, la inicial del nombre más el apellido, la fecha de nacimiento, etc.).
2. Logra entrar a otra red. Desde allí "espía[6]" donde lee el tráfico de mensajes entre esa red y la nuestra y descubre la contraseña que usa alguien o el blanco que se conecta a nuestra red desde la que "invadió". Para ello corre algún programa que le permita analizar todo el tráfico de la red donde está, a la espera de una conexión con otra PC o servidor.
3. En general una vez que está dentro de una red le resulta fácil acceder y puede realizar suplantación de maquinas, ingreso como otro individuo, y ya teniendo una cuenta en el mismo servidor, puede ejecutar programas que realizan la obtención del password débiles e inseguros de los cuales utilizan 2 técnicas:
   - Por diccionario: Es el uso de comparaciones del password cifrado con diccionarios adquiridos en la red.
   - Por fuerza bruta: Donde el intruso corre determinado programa intentado de cualquier forma ingresar por medio de *scripts*.

2.9.2. Uso de vulnerabilidades

A qué se le considera vulnerabilidad. A un error de software o drivers (programación) o de hardware (ejemplo interrupciones), que son encontrados por personal, experto o no. Normalmente encuentran la falla por error al usario, y tratan de obtener ventajas de ellas, como la obtención de una cuenta o el ingreso al sistema. Por omisión Unix entrega el sistema total al

[5]Ingenieria social: Son prácticas comunes, donde averiguan datos esenciales del entorno de trabajo, un ejemplo: el nombre de la mascota del encargado del sistemas

[6]Para realizar el espionaje utilizan los programas como *sniffers*

administrador que tiene permisos absolutos, o al que haya realizado la ejecución pensando que así sera restablecido con éxito.

Estas vulnerabilidad son encontradas y publicadas para su solución. Pero un cierto grupo de la población de Internet, trata de manipular lar las vulnerabilidades de un sistema hasta adueñarse de el sistema, esto dependera del atacante el motivo y el mal uso que pueda ocasionar.

La mayoría de los ataques con éxito provenientes de Internet se pueden agrupar así la utilización de un reducido número de vulnerabilidades. La mayoria de las computadora o servidores comprometidos durante incidentes de seguridad son atacados mediante una vulnerabilidad concreta. Una vulnerabilidad común es el servidor de correo o el servidor de DNS esto es, para ser utilizados para ataques distribuidos de negación de servicio.

Vulnerabilidades críticas

Se pueden encontran información concreta en los siguientes portales como por ejemplo: **SANS Institute y el National Infraestructure Protection Center (NIPC)**, la lista está avalada por expertos de seguridad como NSA, CCERT , FBI, CIAC y SANS, de forma breve colocan las 10 vulnerabilidades más críticas.

Las referencias están clasificadas en CAN y CVE para mayor referencia visite el proyecto de `http://cve.mitre.org` relacionado con `common vulnerabilites and exposures`. Bob Todd autor de la herramienta SARA (Security Auditor's Research Assistant) realizó una herramienta basándose en la lista publicada en SANS/FBI y se puede encontrar en `http://www.cisecurity.org`. Toda liga a la base de vulnerabilidades de ICAT contiene una referencia de enlace a la base de CVE, dicha base esta localizada en `http://icat.nist.gov`. A partir de Octubre del 2001, se clasifican en:

- **VG** Vulnerabilidad general
- **VU** Vulnerabilidad Unix
- **VW** Vulnerabilidad Windows

Solo se señalaran las referentes a la general y Unix.

VG

Instalación de sistemas por default.

Cuentas de usuarios sin passwords o passwords débiles.

Respaldos inexistentes o Incompletos.

Puertos abiertos(de servicios de red, no usados o vulnerables).

Uso inapropiado de filtrado de paquetes nulos.

Registros incompletos o no existentes.

Programas CGI Vulnerables|Sólo en los casos de que existan páginas web.

VU

Buffer Overflow's[7] en servicios RPC[8]

Vulnerabilidad en Sendmail[9] Sólo se aplica a aquellos que tienen servidor de correo sendmail.

Vulnerabilidades en BIND[10] Solo se asumen a redes tipo B con DNS.

Ya fueron mencionadas las vulnerabilidades que realizan una búsqueda exaustiva en sitios de la red, donde obtienen programas llamados *exploits*, el *exploit* ejecuta la desventaja del sistema y entrega así al individuo que generó el acceso parcial o total del servidor. Por ejemplo la vulnerabilidad de lpr, donde el servidor de impresión ultiza un puerto de comunicación, sin olvidar el reciente problema de las estaciones de trabajo `SUN` donde la administracón remota siempre está activa en la instalación por default, o el común sunrpc.

Un grupo de atacantes experimentados ya realizan gusanos como `Ramen`, `Nimda`, `Code Red`, `etc.`, que usan este TOP para realizar un numero mayor de intrusiones en menos tiempo y de manera automatica.

2.9.3. Buffer Overflow

En la subsección anterior se tocó este tema pero en realidad no se ha explicado qué es y cómo funciona. Es un error, del más comun, y sin duda el más conocido y utilizado es el *stack smashing* o desbordamiento de pila, también conocido como *buffer overflow*.

Su origen es por el mal hábito al programar; los programadores no son adaptados a que sean defensivos. Este problema está presente en múltiples programas, al menos controlados. Aun se

ven con frecuencia alertas sobre programas que se ven afectados por desbordamientos. La idea del *stack smashing* es sencilla: en algunas implementaciones de C es posible corromper la pila de ejecución de un programa escribiendo más alla de los limites de un array declarado auto en una función; esto puede causar que la direccion de retorno de dicha función sea una dirección aleatoria. Esto, unido a permisos de los archivos ejecutables en Unix (principalmente a los bits de SetUID y SetGID), hacen que el sistema operativo otorgue acceso root a usuarios sin privilegios.

Por ejemplo, imaginemos una función que trate de copiar con strcpy() un array de 200 caracteres en uno de 20: al ejecutar el programa, se generará una violación de segmento y por tanto el clásico *core dump* al que los usuarios de Unix estamos acostumbrados. Se ha producido una sobreescritura de la dirección de retorno de la función; si logramos que esta sobre escritura no sea aleatoria sino que apunte a un código concreto (habitualmente el codigo de un *shell*), dicho código se va a ejecutar.

El problema reside en los archivos con privilegios *setuID* y *setgID*; recordemos que cuando alguien los ejecuta, está trabajando con los privilegios de quien los creó, y todo lo que ejecute lo hace con esos privilegios incluido el código que se ha insertado en la dirección de retorno de nuestra función problemática. Si como hemos dicho, este codigo es el de un intérprete de comandos y el archivo pertenece al administrador, el atacante consigue ejecutar un shell con privilegios de root.

```
-rwsr-xr-x  1  root  root  20120  jun  25  2001  traceroute
```

Existen multitud de exploits (programas que aprovechan un error en otro programa para violar la seguridad del sistema) disponibles en Internet, para casi todas las variantes de Unix y que incluyen el código necesario para ejecutar shells sobre cualquier sistema operativo y arquitectura.

Para minimizar el impacto que pueden causar los desbordamientos en nuestro sistema es necesaria una colaboración entre fabricantes, administradores y programadores. Los primeros han de tratar de verificar más la robustez de los programas críticos antes de distribuirlos, mientras que los administradores han de mantener al mínimo el número de archivos en sus sistemas y los programadores tienen que esforzarse en generar cóodigo con menos puntos de desbordamiento.

Los sitios de dónde obtienen la información son muchos, va a depender de las condiciones tanto económicas, sociales, pero todos tienen en común usar la perturbación de sistemas ya sea por curiosidad o por malicia. Se muestra una lista de lo que un *hacker* conoce y considera necesario para satisfacer las necesidades de nuestros días.

Ejemplos

Con el objetivo de infectar el mayor número de computadoras posibles, cada vez son más los virus que recurren a la Ingeniería Social, habitualmente empleada por los hackers para engañar a los usuarios A juzgar por los últimos casos, entre los que destaca el reciente HomePage, los usuarios siguen cayendo víctima de esta técnica. Por este motivo, Panda Software, compañía dedicada al desarrollo de antivirus, ha lanzado un comunicado en el que explica el funcionamiento de la ingeniería social y la manera de evitar ser engañado.

En la Ingeniería Social no se emplea ningún programa de software o elemento de hardware, sólo grandes dosis de ingenio, sutileza y persuasión para así lograr datos de otra persona sin que se dé cuenta de que está revelando información importante con la que, además, el atacante puede dañar su computadora.

Un claro ejemplo de Ingeniería Social es el del *hacker*que llama por teléfono a una empresa para decir que necesita ayuda o hablar con el administrador de la red porque hay que modificar algún aspecto de la configuración.

Durante la conversación, y a través de escogidas y cuidadas preguntas, el atacante obtendrá los datos (como los códigos de acceso a los equipos) que necesita para vulnerar la seguridad del sistema. En la práctica, los autores de virus emplean la Ingeniería Social para que sus creaciones se propaguen rápidamente.

Para ello atraen la atención del usuario y consiguen que realice alguna acción (que, normalmente, consiste en abrir un archivo que es el que procede a realizar la infección), mediante variados trucos, entre los que destacan los siguientes:

- Emplear como señuelos mensajes o archivos con explícitas referencias eróticas
- Aludir a personajes famosos, tal y como ha sucedido con Anna Kournikova famosa tenista dedicada al show bussines
- Servirse de "ganchos" vinculados a las relaciones amorosas

Algunos virus son scripts para HTML y sobreponen una ventana HTML encima de la que muestra el navegador preguntando si desea o no ejecutar el código que lleva consigo la página web. En realidad, la ventana que presenta el virus oculta la alarma y presenta un mensaje que incita al usuario a elegir una opción que permitirá al virus infectar el sistema.

De lo anterior sacamos dos conclusiones muy importantes:

1. La primer barrera, y la más difícil de sortear, para ingresar a un sistema, la conforman las contraseñas de los usuarios de la red (la de cada uno). No sirve que 10 o 20 tengamos contraseñas muy sofisticadas si los otros 300 usuarios están "regalando"el acceso a nuestra red.

2. Se puede descubrir una contraseña espiando desde otra red. Por segura que sea la nuestra, si en el resto no se toman iguales medidas el problema subsiste, y lo único que podemos hacer es cambiar con cierta frecuencia las contraseñas de los usuarios. Al respecto debe tenerse especial cuidado cuando se le asigna a alguien acceso a más de una red. Un uso irresponsable en una red compromete la seguridad de las otras.

Sobre cómo alguien aprende todo esto, podemos decir que aparte de la bibliografía por ellos generada y las horas dedicadas al tema pueden comenzar leyendo documentos donde se explica a los usuarios de una red sobre qué medidas de seguridad deben adoptar.

Qué es lo que pueden obtener de mi sistema. Depende del atacante, ya que varia su modo de operar. Cada uno quiere llevarse un tipo de información o modificar datos por dar un ejemplo, puede modificar datos de nómina, o la página web del sitio, o la instalación de un chat y después difundirlo en los canales del hackers dentro del Mirc, o aparecer en http://www.atrition.com, o en http://www.cnnespanol.com, o en la portada del New York Times.

2.9.4. Barrido de puertos

Es el rastreo de puertos(`Scanner`), es una herramienta que busca los puertos que se encuentran esperando una conexion al sistema.

La idea principal de un rastreador es buscar un objetivo y a través del envio de tramas que pueden ser tramas no válidas y tramas válidas, buscar por servicios activos y puertas abiertas en todos los sistemas, para determinar si existe alguna vulnerabilidad explotable remotamente para poder accesar al sistema.

Un barrido de puertos por lo regular no deja rastros en el sistema ya que existen diversos métodos para fragmentar los paquetes TCP.

Técnicas de los Barridos de Puertos Cada máquina en Internet cuenta con una dirección IP que es única, pero al pedirle la petición de una señal sincrónica nos puede responder de formas distintas cada puerto de comunicación. El intruso averigua direcciones de posibles

blancos, otros los buscan circunstancialmente o por su nombre canónico, recordemos que no únicamente nos enfrentamos a inexpertos, localizan si existe un firewall, si las máquinas tienen un alto índice de penetración. Los puertos representan un potencial canal de comunicación en el cual los protocolos TCP/UDP/ICMP juegan papeles importantes. Los métodos más comúnmente utilizados son los siguientes:

- Vanilla TCP connect().
- TCP SYN(half open)
- TCP FIN(stealth)
- TCP FULL (connection)
- SYN/FIN(Usando tiny Fragmentation)
- UDP(recvfrom())
- UDP raw ICMP
- ICMP Packet.
- Reverse ident

Existen varios métodos para el barrido de puertos realizados por los intrusos, mediante la técnica StackFingerprint, la cual a través del tamaño de tramas se puede determinar el tipo de Sistema Operativo, facilitando información potencial para los intrusos.

2.9.5. Herramientas Utilizadas

Va a depender de las circunstancias o el objetivo, todos deben de ser ejecutados por el administrador o con privileguios similares, exceptuando nessus que es un programa mejor elaborado y permite que existan agentes de seguridad. Cada uno de estos programas son de uso administrativo aunque son muy recomendados en listas de script kiddies, las herramientas de uso comunitario son:

saint

NMAP

z0ne

cracker.pl

queSo

satan

MSCAN

SSCAN

Nessus

Y para uso particular en una gama de servicios que por omisión tienen alto riesgo de vulnerabilidad como por ejemplo `DNS, ftp, rpc`. Por ejemplo:

Rastreadores de DNS

Rastreadores de Snmpdix

Rastreadores de wu-ftpd

Rastreadores de rpc.statd

En las bitacoras del sistema se muestra de la siguiente manera.

```
\# less /var/adm/messages
 Ene 1 17:01:37 6D:victima ftpd[1038976]: connection from atacante.org
 Ene 1 17:11:07 6D:victima telnetd[1046061]: connection from atacante.org
 Ene 2 17:01:37 6Q:victima sshd[1038976]: connect from atacante.org
 Ene 3 17:01:37 6D:victima rlogind[1038976]: connection from atacante.org
```

Como podemos observar el patrón es muy distinto solicitando diversos servicios en la misma hora con un desfase en los minutos por la lectura de puertos no sólo las realizan atacantes también son responsabilidad y ética de los administradores, y este acto debe de estar especificado en las políticas de seguridad.

Servicios de red activos.

Cada servicio que fue detectado es porque en la instalación del servidor fue activada, o necesaria para su funcionamiento. Los puertos residen en /etc/service y cada servicio puede ser activado o desactivado dependiendo del inicio de sesión. Por ejemplo INIT 1 es conocido como singles user, esto es que desactiva todos los servicios de red, y sólo está el administrador, INIT 3 es el modo en que levanta el sistema, en modo texto e implica que el sistema y servicios de red sean activos, INIT 5 donde equipos IRIS, SOLARIS, AIX, hasta LINUX puedan ingresar en modo X (modo de ventanas). Cada uno de estos se puede revisar y adjudicar los servicios en /etc/rc.d/rcX donde X es el valor de INIT para el servidor, en dicha carpeta se encuentran cada servicio con un valor numérico y una letra S o K, la S es que va a levantar ese servicio y la K que no le va ha hacer caso o matar en caso de haberse levántado. Esto es usado si un atacante modifica un binario si reinicia el sistema y levanta un binario que quiera o hasta el kernell.

Que tan importante es que nos conozcan y los servicios que ofrecemos. Con la información de los puertos activos, sistema operativo y kernel pueden buscar en sitios bien conosidos de exploits, y detonar alguno que corresponda con las caracteristicas de nuestro servidor, ya sea es explote remotamente o local, a su vez puede obtener privilegios de root donde inflinge trabajando en el sistema obteniendo las tablas de password que se ubican en /etc/password y la de shadow localizada en la misma carpeta de /etc ; coloca caballos de troya o bombas lógicas, puede leer correos electrónicos, suplantación de identidad, buscar rhost shost, saber quiénes son los hospedados en este servidor, sin olvidar usar, manipular, o destruir datos y/ o modificación de ellos.

Dentro de los ataques más frecuentes se encuentran los llamados " troyanos", que en recuerdo al Caballo de Troya se introducen en la computadora al ejecutar otro programa, tales como el envío de un correo electrónico. Algunos de los programas que se utilizan para combatirlos están dentro de la web, pero muchos de ellos son virus creados por los propios hackers para evitar la destrucción del primero. El mercado, para evitar semejantes problemas, ha lanzado nuevos productos "antivirus" que luchan incluso contra los más difíciles de exterminar.

Otro tipo de virus son las "WEB maliciosas", que como su propio nombre indica son páginas web con contenidos maliciosos, las cuales aprovechan la posibilidad de hacer "pequeños programitas" soportados por el estandar HTML, que son realizados en JavaScript, VbScript, ActiveX, Apple de Java y el uso de Cookies.

Los "Programas defectuosos.aprovechan los defectos de seguridad y hacen que el computadora se quede bloqueada en sus dispositivos como el teclado, el ratón y demás accesorios de entrada

de datos hacia el computadora; también pueden apagar la conexión a la red u originar otros ataques que dependerán de la seguridad del programa que el usuario utilice.

Los expertos en virus y piratas informáticos aconsejan que los usuarios eviten grabar el password de acceso a Internet en el disco duro; si el *hacker* intenta averiguar el password puede tardar horas dependiendo de la capasidad de cómputo.

Dentro de las llamadas "hacked homepages" cualquier usuario de Internet puede encontrar una lista detallada de las distintas Websites que han sido víctima de un virus informático. Esta lista indica la empresa, la fecha, las observaciones e incluso en algunos casos los causantes del problema. En ella se encuentran grupos como la FOX, que emitió el programa "Expediente X" durante varias horas por causa del virus; la NASA que fue "hackeada" en dos ocasiones, e incluso el servicio de seguridad de la CIA, entre otros muchos.

2.10. Reingreso al sistema y sus mecanismos

Todo atacante desea reingresar a un equipo ya rota la seguridad del sitio, para ello crea las puertas traseras por las cuales puede reingresar las veces que quiera, para esto, existen otros métodos como el manejo de complejos programas llamados rootkit que pueden deshabilitar cierta rigidez en la infraestructura de seguridad y administración, me refiero, a la importancia de las bitácoras, donde cada suceso es descrito, pero tal programa puede borrarlo o modificarlo conjunto a una serie de binarios como el de (w, who, ls,ps, entre otros comados basicos).

Cuando logran el objetivo de ingresar colocan programas como rootkits, backdoor, esto para poder burlar a la posible administración del sistema, y así poder reingresar a `victima.org` en futuras ocasiones. Los lugares comunes donde son ubicados son archivos del sistema son /bin ;/usr/bin ;/usr/sbin. Estos están en carpetas ocultas, y al instalarlos abren puertos de comunicación dell servidor hacia el exterior.

2.10.1. Códigos maliciosos

Los códigos maliciosos son programas el cual se ejecuta en su nivel más primitivo, bajo línea de comandos que modifican o alteran la estructura natural del sistema, entre estos códigos se hace mención de virus que aunque para Unix no se les considere asi, Fred Cohen en 1983 creó un shell scripts que ejecutaba con éxito. Aunque en Unix es común escuchar de los

exploit; los cuales son entregados como un obsequio llamado así troyano.

Otra variante es el conocido gusano, que adquirió su nombe en 1975 en la obra de ciencia ficción de John Brunner "The Shockwave Rider" hace referencia a programas capaces de viajar por sí mismos a través de redes de computadoras, una vez alcanzada una máquina; donde el gusano pueden instalar en el sistema alcanzado como un virus, atacar y la manera de localización es el increible ancho de banda en la red afectada. Recordando historicamente el gusano de Worm en 1988, previamente Bob Thomas generó CREEPER un software que no se le consideró un programa malicioso porque invadía la integridad de un avión y se la notificaba al aeropuerto para la obtencióon de los datos de control de vuelo. Pero esto ya se habia iniciado desde 1980 en Xerox en Palo Alto California, donde sus desarrolladores John Shoch y Jon Hupp, se dedicaron a tareas como el intercambio de mensajes entre sistemas o el aprovechamiento de recursos ociosos durante la noche. Pero el programa se desbordó eso dio el origen a una cura para este virus, donde no sólo estaba funcionando en tiempos muertos sino también en procesos de alto rendimento como lo era en la mañana y tarde. Este es considerado el primer incidente de seguridad en el que entraban worms en juego[11].

Conejos

Los conejos o bacterias son programas que de forma directa no dañan al sistema operativo, solo se reproducen, generalmente de forma exponencial, hasta que la cantidad de recursos consumidos (procesador, memoria, disco, etc.) se convierte en una negación de servicio para el sistema afectado; esto es si no tienen quotas[12] o una administración de los recursos.

2.10.2. Caballos de Troya

Su nombre proveninte de la Odisea de Homero donde se narra la lucha entre Griegos y Troyanos por la posesión de Helena. Los griegos decidierón dejar un gran caballo de madera como obsequio en la puerta de la ciudad, en su interior se escondieron soldados y cuándo menos lo esperaron, salierón a conquistar la ciudad y reclamar todo como propiedad de grecia.

Comptacionalmente, un Caballo de Troya es un programa que realiza alguna acción no

[11]Para mayor referencia revie John F. Shoch and Jon A. Hupp. The worm programs - early experience with a distributed computation. Communications of the ACM, 25(3):172-180, Marzo 1982.

[12]la asignación de quotas es declarar que cada usuario va a utilizar un espacio maximo en disco duro y no sobrepasarse de la cantidad asignada

documentada que el programador ha decidido llevar a cabo, pero que el usuario rechazaría si estuviera al corriente de ella. Según algunos, un virus constituye un caso particular de Caballo de Troya, es decir un virus capaz de propagarse a otros programas (los convierte también en troyanos). Según otros, un virus que no provoca ningún daño deliberadamente (salvo la replicación) no es un troyano. Por último, a pesar de las definiciones, muchas personas utilizan el término "troyano"para referirse únicamente a un programa nocivo "no replicador", a modo de distinción entre el conjunto de troyanos y el conjunto de virus.

Existen 2 tipos de caballos de Troya los inofensivos que desafortundamente no abundan, y solo dan alerta al administrador de un suceso de malicia; y los dañinos que se adjudican el nombre : trojan mule o mula de Troya; es típico encontrar el falso programa de login.

2.10.3. Applets hostiles

En los ultimos años, donde el web, Java y Javascript ha crecido, una nueva forma de software se ha hecho popular. Se trata de los denominados applets hostiles, applets que al ser descargados intentan monopolizar o explotar los recursos del sistema de una forma inapropiada, con el solo hecho de visitar una paguina web. Desde un simple ataque de negacion de servicio o ejecución remota de exploits hasta prácticas elaboradas, como difusion de virus, ruptura lógica de cortafuegos.

Hoy en dia Sun Microsystems trata de minimizar los efectos potenciales de estos applets; principalmente se han centrado esfuerzos en controlar la cantidad de recursos consumidos por un programa y en proporcionar las clases necesarias para que los propios navegadores monitorean los applets.

2.10.4. Bombas lógicas

Son muy parecidas a los troyanos; se trata de código insertado en un programa que parece realizar cierta acción útil. Pero mientras que un troyano se ejecuta cada vez que se ejecuta el programa que lo contiene, una bomba lógica sólo se activa bajo ciertas condiciones, como una determinada fecha, la existencia de un archivo con un nombre dado, o el alcance de cierto número de ejecuciones del programa que contiene la bomba; puede permanecer inactiva en el sistema durante mucho tiempo sin activarse y por tanto sin que nadie note un funcionamiento anómalo hasta que el daño producido por la bomba ya está hecho. Este es un claro ejemplo de un ataque pasivo.

2.10.5. Los rootkits

Son un conjunto de caballos de troya o shellscript que modifican los binarios o programas del sistema como por ejemplo (netstat[13],ps [14], lsof[15], w[16], . . .), para conseguir que cuando el administrador las ejecute no vea la información relativa al atacante, algunos están tan bien elaborados que al utilizar herramientas de auditoría estos no detectan cambio en sus estructura pero generando los md5 o las llaves de con `tripwire`[17], se detecta de inmediato, como sus procesos o su conexión al sistema. Otro programa que se suele suplantar es login, por ejemplo para que al recibir un cierto nombre de usuario y contraseña proporcione acceso al sistema sin necesidad de consultar `/etc/passwd`. Y que en futura ocasión regrese el atacante a través de "puertas traseras" (backdoor's) esto a fin de asegurar sus reingresos al sistema atacado. Este rootkit introduce a escondidas una implementación que va substituir un programa de la distribución o de terceros donde realiza las actividades que el usuario espera que haga más aparte otro tipo de acciones.

2.10.6. Sniffers

Para poder hablar de `sniffers` se debe de hablar de redes. La mayoría de las redes locales actuales utilizan como nivel de enlace alguna de las variantes de ETHERNET. Las primeras versiones de Ethernet utilizaban un único cable coaxial que unía a todas las máquinas.

Cualquier paquete que se transmite por el cable es escuchado por todas las máquinas (la tarjeta de red). Aunque la tarjeta de red puede leer todos los paquetes que circulan por la red, normalmente están programadas para aceptar sólo aquellos paquetes cuyo destino es la máquina local.

Las nuevas versiones de Ethernet basadas en cables de par trenzado 10BASET y `hubs`, que utilizan una topología en estrella, siguen conservando el esquema de funcionamiento original de difusiones globales. Existen hubs, `switches` con capacidad de aprendizaje y auto configuración que son capaces de limitar la difusión de los paquetes sólo a aquellos segmentos donde se encuentre la máquina destino.

Todas las tarjetas de red se pueden programar para que escuchen todos los paquetes que

[13]netstat: monitor de aplicaciones de red
[14]ps: visor de procesos a ejecución
[15]lsof: nos es útil para el análisis de qué y quién está ejecutando un archivo o programa
[16]w: revisa la variable login activa
[17]tripwire: utileria práctica para la revisión de la integridad del sistema

pasan por la red. Este modo de operación se conoce como modo `promiscuo`.

El `sniffer` es una aplicación que pone la tarjeta de red en modo promiscuo para poder leer todos los paquetes. Normalmente el sniffer es capaz de filtrar y ordenar los paquetes para obtener un resultado comprensible. En Unix es necesario ser root para poder poner la tarjeta en modo promiscuo. Con la orden `ifconfig` se puede conocer el estado de las tarjetas de red y modificarla.De tal manera, al ver todas las cadenas que estan pasando por la tarjeta de red, se monitorea cualquier suceso.

Las herrramientas son varidas, daré algunos ejemplos en el capitulo de reformas en la seccion de herramientas.

2.10.7. Spoofing suplantación de identidad

Es un ataque que se autentifica, notificando que es una máquina por otra, mediante la falsificación de paquetes de un `host` que es de confiánza. Los ataques de `spoofing` se dividen:

Por IP, donde el servidor que va a otorgar el axeso restringuido a un grupo de máquinas y los usuarios se comunican de metodos de confiánza por ejemplo rlogin, rsh, rcp, rcmd sino también al comunicarse de "servidor.victima" con "servidor.atacante" debe conocer la secuencia sucesiva de los paquetes TCP, se escucha sencillo pero el "servidor.atacante" no escucha la pregunta de petición de "servidor.victima" así que la secuencia complica el ataque. Hoy en dia es comun en sitios de confianza, como los son, los clusters, redes privadas, etc.

Por ARP. Es una pequeña variación de confianza, donde se verifica el protocolo de resolución de direcciones, donde se verifica la dirección del hardware y de la IP.

A través DNS, donde el principal manjar es el servidor de DNS y modifica de forma explícita las tablas de dirección de IP.

Últimamente han aparecido en segumiento poe correo acerca del tema de los abusos del protocolo DNS. DNS es el protocolo que se usa para convertir IPs numéricas en Fully Qualified Domain Name (FQDN)[18] y al revés. El funcionamiento de un servidor DNS (Nameserver - NS) es la de resolver IPs a nombres y al revés.

[18]FQDN: Si deseas mayor información consulta los RFCs que tratan todos los aspectos de DNS: 1031, 1032, 1033, 1034, 1035

Nuestro DNS es dns1.unam.mx y el nombre que nos han dicho que resolvamos es "ataque.victima.unam.mx Lo que hará nuestro nameserver es desglozar la dirección en partes. Primero consultará a uno de los ROOT-nameservers de Inet quien sirve el dominio "mx". El DNS tiene una lista de estos ROOT-DNS con sus IPs asociadas, luego no tiene que consultar ya que IPs tienen (no conseguiria resolver nada). Bueno, pues consulta el dominio "mx" . Entonces obtiene la dirección del DNS que sirve al dominio "mx". A continuación pregunta a ese DNS quien sirve al dominio "unam" . Obtiene la dirección de otro DNS, a éste le pregunta quién sirve el dominio "victima", así obtiene la dirección de otro DNS y le pregunta qué IP tiene la maquina .ªtaque". Asi, ya ha obtenido la IP de "ataque.victima.unam.mx". Pongamos, por ejemplo, que es 132.248.123.123.

Aquí ha terminado el trabajo de nuestro DNS, que además de resolver la dirección, la guarda en su cache para futuras consultas. La cache es un registro de las últimas direcciones que resolvió el DNS, por si se le consulta sobre éstas no tiene que volver a resolverlas.

Nota: Éste es el funcionamiento de un DNS que soporta la función de recursividad (la mayoria). Esto significa que si el DNS no proporciona la dirección que se le pide que resuelva, enviará la petición a otro DNS para que la resuelva, hasta obtener la resolución o un error de dirección irresoluble. Si el DNS no tiene esta función de recursividad, simplemente, resolverá el dominio de la dirección que le pedimos que resuelva la resolvera. Por tanto, las vulnerabilidades que se explican más adelante, SÓLO AFECTAN A SERVIDORES DNS QUE SOPORTAN RECURSIÓN.

Datagramas DNS

El protocolo DNS se comunica por UDP, un protocolo de transporte sin conexión, es decir, que lanza el paquete a la red, y ya no se preocupa más de él. Ésto imposibilita que se lleve un control del flujo de la conexión, corrección de errores, etc. Éste es un punto muy importante, deviado a UDP no puede, al contrario que TCP, mantener un flujo de comunicación entre dos hosts, con los ya conocidos sequenc numbers.

Así, si el atacante quiere realizar spoofing o meternos en medio de una conexión UDP no tendremos que predecir estos seqnums[19].

Dado que el protocolo de transporte no identifica los paquetes por conexión, debe ser el protocolo DNS el que asigne a los paquetes una " identificación" para poder saber qué paquete

[19]Si deseas conocer más de la estructura de los datagramas del DNS en la RFC 1035 muestra los datos más concretos sobre cómo funciona.

responde a una pregunta anterior, etc. A esta identificación se le llama QueryID y asigna el DNS que inicia la comunicación (el que pregunta). A partir de ese momento todos los paquetes referentes a la resolución de esa pregunta iran identificados con ese QueryID.

Inyección de datos falsos en la cache de un DNS

A continuación viene la parte interesante, cómo engañar a un DNS para que resuelva un nombre dado a una IP que queramos nosotros o viceversa.

Método A: Suplantación de un DNS. Para este ataque debemos disponer de:

- Un DNS primario (al que pueda consultar cualquier computadora sobre un dominio)

La técnica es la siguiente. Digamos que nosotros somos m1.hh.org (1.1.1.1) y el ns que controlamos es ns.xs.com y queremos spoofear como ip.dead.org al DNS ns.victim.com.

Se trata de hacernos pasar por el ns de dead.org y hacer creer a ns.victim.com que ha resuelto bien a ip.dead.org. Para crear este "paqueteçon información falsa, el DNS de victim.com debe preguntarnos sobre él, y será en ese momento cuando nosotros contestemos. Pero como no somos el DNS de dead.org no podemos saber cuál será el QueryID de ns.victim.com. Por tanto tenemos un problema, ¿Como sabemos el QueryID que tendrá el paquete que preguntará sobre ip.dead.org ?

Si hacemos una consulta a dns.victim.com sobre una dirección que sirva el dns.xs.com (por ejemplo, consultamos www.xs.com), dns.victim.com se pondrá en contacto con dns.xs.com y le preguntará la dirección de www.xs.com. nuestro DNS le responderá, y tan contentos. Dado que previamente habremos colocado un sniffer la conexión con nuestro DNS por parte de dns.victim.com preguntando por la dirección de www, sabremos con que QueryID ha preguntado.

El QueryID del protocolo DNs no es un número aleatorio o pseudo-aleatorio como el de algunas implementaciones de TCP, sino que es secuencial, ésto es, que para cada pregunta, aumenta en uno el QueryID. Esto es así porque el QueryID no se diseñó como un mecanismo de seguridad ante posibles spoof's, sino como una manera de controlar que respuesta corresponde a que pregunta y viceversa.

Una vez que tenemos el QueryID del DNS víctima creamos un datagrama DNS con la información falsa que queremos transmitir (ip.dead.org ¡-¿1.1.1.1), como si la enviara el DNS de dead.org, y con destino a dns.victim.com. Hacemos una consulta al

dns.victim.com preguntando por alguna dirección del dominio dead.org y inmediatamente enviamos el paquete de respuesta spoofing. En lugar de enviar un paquete es conveniente enviar unos cuantos, con QueryIDs consecutivos, dado que en el tiempo que hemos tardado entre que obtenemos el queryID y solicitamos la consulta por lechuck.org el DNS victima puede haber recibido otras peticiones de resolución, y por tanto el QueryID puede haber aumentado ligeramente.

Así, si dns.victim.com recibe antes nuestra respuesta que la respuesta del DNS de lechuck.org (en caso de que exista) o un error indicando que no hay tal dominio (en caso de que no exista), ya tendremos en la cache del DNS víctima la información falsa que queriamos inyectarle.

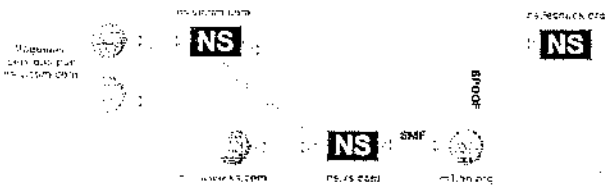


Figura 2.1: Abuso de DNS

Método B: Inyección de datos en nuestra respuesta. Para este ataque debemos disponer igual que antes de un DNS primario.

Cuando un DNS hace una consulta a otro no sabe que numero de respuestas puede obtener a su busqueda, ni le importa, pues todo lo que sea información lo recibirá con mucho gusto. Pues bien, si al hacer una consulta a un NS, además (o en lugar) de la información que solicita se le devuelve la informacion falsa que nosotros queramos, el DNS que preguntó la aceptará y ya la tendremos en la cache del DNS que nos preguntó.

Aplicado al caso anterior, lo único que debemos hacer es preparar nuestro DNS (dns.xs.com) para que responda con la información falsa

```
(ip.dead.org <->1.1.1.1)
```

a las preguntas de un DNS (dns.victim.com) y hacer que el DNS víctima pregunte a nuestro DNS.

Este método es mucho más sencillo y efectivo que el anterior, aunque requiere mas recursos para que funcione, pues debemos modificar el programa servidor de nombres o bien hacer un programa que funcione como tal, y que sirva nuestros datos falsos.

textbfImpacto.

Aparte del uso que todos pueden estar pensando para este tipo de ataques a DNS (fanfarronear en IRC) hay multitud de usos mucho más serios, en que la seguridad de muchos sistemas puede quedar comprometida:

- NFS (Network File Sharing) en sistemas que exporten para nombres de hosts
- Servicios r* (rlogin, rsh, etc.) haciéndonos pasar por "trusted hosts".
- TCP Wrappers que se basen en nombres de hosts para cortar el paso.
- entre otros mas.

Supongase que ya tenemos lo que necesitabamos, un host que sea DNS primario de un dominio y otro host donde haremos correr nuesto programa "inyector" de información falsa DNS. Si se tubiera la situación siguiente:

- Nameserver primario: dns.xs.com
- Host que corre el inyector: 123.123.123.123

Expondre cuales serían las modificaciones que deberan hacerse a la configuración del demonio DNS suponiendo que éste es el "named", que con mas probabilidad se encontrará en un server unix.

Se consultar en el archivo de configuracion del named donde almacena la informacion de los dominios que sirve. Este archivo es el /etc/named.boot (aunque en las últimas versiones de bind, se ha trasladado a /etc/named.conf y se ha modificado un poco la estructura del archivo, en cuyo caso se tendra que adaptar en la presente explicación un poco). En nuestro ejemplo, contiene lo siguiente:

```
                        cache       .           root.cache
directorio /var/named   primary     xs.com      db.xs.com
                        secondary   robin.org   db.robin.org
```

Analizando línea por línea: directorio /var/named : Directorio base donde almacena los archivos de configuracion de los dominios primary xs.com db.xs.com : Es un DNS primario del dominio xs.com, y lo almacena en el archivo db.xs.com (Este es el que nos interesa) secondary robin.org : Es DNS secundario del dominio robin.org, y lo almacena en el archivo db.robin.org (Este no nos interesa) Observando al archivo db.xs.com (que como sabemos, estara en /var/named) encontramos que:

```
xs.com.  IN  SOA  dns.xs.com.  root.xs.com. (
                                1997062100
                                28800
                                7200
                                604800
                                86400)
```

```
xs.com.          IN  DNS  dns.xs.com.
localhost        IN  A    127.0.0.1
XXX              IN  A    10.0.0.1
elaine.xs.com.   IN  A    10.0.0.2
```

Las primeras líneas se refieren a la configuración del dominio, quen gestiona las fechas de expiracion, etc. A continuación está la configuración de los Nameservers del dominio: xs.com. IN DNS dns.xs.com. En esta línea se especifica que el host "dns.xs.com" sirve al dominio xs.com.

Aquí es donde vamos a hacer la modificación: Tenemos que insertar una línea que cree un subdominio y que ponga a nuestro host inyector como nameserver de ese subdominio. A la derecha de un IN DNS sólo se acepta un FQDN, es decir, no podemos poner una ip numerica. Así que vamos a tener que crear un host y luego asignarle una IP. La primera modificación, quedaría así con lo que hemos supuesto antes:

```
xs.com.      IN  DNS  dns.xs.com.
sub.xs.com.  IN  DNS  ns1.xs.com.
```

Ahora tendremos que asignarle la IP a este host nuevo que hemos creado (ns1.xs.com). Para asignar una IP numérica a un FQDN, se usa el RR "IN A", tal como se puede ver en las siguientes líneas:

```
XXX              IN  A  10.0.0.1
elaine.xs.com.   IN  A  10.0.0.2
```

La segunda línea, asigna la IP 10.0.0.2 al FQDN pitufa.xs.com. El punto al final de pitufa.xs.com. es muy importante, ya que si no le ponemos punto (como a XXX) el named le pone al final el nombre del dominio (xs.com).

Asi, para insertar la línea que queremos, y segun lo que hemos supuesto antes, utilizariamos esta línea:

```
ns1.xs.com.  IN  A  123.123.123.123
```

Asi ya tenemos el named configurado para el spoof. Solo nos queda recargar esta informacion en el named (killall -HUP named), esperar a que la informacion se propague y poner a trabajar el programa inyector.

Como programa inyector, utilisan el jizz (fuente en .c).

La informacion mostrada es un panorama general de la metodologia de los atacantes, pero cada individuo y su futura victima va ha crear un entorno distinto, sin olvidar que estas tecnicas van a ir evolucionando, solo se mostro las bases o definiciones. Esta carrera de la administración conjunto a la seguridad, quien va ha ganar es la madures de la computación.

Si se preguntan en donde obtienen la información de ataques revise el anexo de referencias de la red.

Capítulo 3

Análisis en un sistema anámalo

En este capitulo denotaré actividades de intrusión a un sistema UNIX durante y después de intentos de intrusión. Tomando en cuénta qué los ataques nunca son iguales, ni tampoco el tipo de plataformas atacadas, ya sea PC's, estaciones de trabajo, servidores cluster, grid o supercomputadoras.

3.1. Siclo de vida

En un componente de seguridad computacional, los incidentes de seguridad es un componente que no se debe de dsplazar, debido al riesgo que se corre día a día. El ciclo se puede dividen en 3.

1. Convivencia: Donde se provee los servicios de red e involucra, protegerse de denegación de servicio, repudio a las comunicaciones. La mejor practica es basado en correr los minismos riesgos.
2. Deteción: Es establecer monitores de culaquier suceso, devidamente vigilados.
3. Respuesta a Incidentes: Son estrategias definidas en las politicas de seguiridad.

El motivo del análisis, es que personal no autorizado ingresa u obtiene privilegios no autorizados. Devido a multiples factores ya sea que no existe una seguridad solida o implicando alguno de los siguientes factores:

- Una mala administración,
- Contamos con un sistema operativo debil,
- Desactualizado para las condiciones optimas,
- No se hizo bien la planeación del sitio,
- No existen politicas de seguridad o no estan siendo cumplidas al pie de la letra.

3.2. Cuando aún se tiene el control del sistema

Recordemos que toda computadora conectada a internet esta propensa a ser atacada la diferencia es cuestion de tiempo y de administración. A los administradores o responsables nos beneficia conocer la seguridad del sistema, conociendo el funcionamiento de éstos para encontrar las debilidades del sistema. Teniend un monitoreo constante de la integridad de el sistema, archivos y servicios que prestan a redes locales como publicas.

Cada acto o acción que está ejecutando UNIX debe de estar controlada por el sistema y cualquier sospecha o anomalia en el sistema se debera analizar por el administrador o responsable; esto por la razón de que puede ser indicador de un ataque ya sea en progreso o que y estemos comprometidos si se presentan actos como la negación de servicio, intentos repetidos de acceso fallidos, cuentas nuevas, archivos modificados, discrepancia en la contabilidad, sesiones sospechosas e imposibles, reportes de usuarios o administradores. Ante estos casos se deben de tomar en cuenta las siguientes consideraciones:

- Definir que es lo qué queremos conocer, si existiera el caso de una infiltración, se necesita saber quién fue o simplemente saber por dónde ingresá y que ocasiona con su visita. Y hasta que punto ocurrio el daño.
- Contamos con los privilegios del administrador, o tenemos el control absoluto del sistema. Para poder ejecutar aplicaciones que se necesitaran.
- En que se va a confiar. Recordemos que si el atacante obtuviera el acceso de administrador podria hacer lo que sea, o de el menor de sus caos el grupo de administracion de sistema podria hacer una tarea menor pero implicaria de impunidad de lo dato que tiene a su cargo.

Se tendrá que revisar los binarios o programas ejecutables, las variables de ambiente ya sea manualmente o con herramientas ya diseñadas para tal acción, por ejemplo utilizando rastreadores o utilerías del sistema nos damos cuenta que el intruso sigue conectado al equipo. Con respecto a esto tenemos el suficiente crirterio para desconectarla de la red o no, el servidor es de misión critica, el valor de la pérdida es muy amplio o se puede recuperar con los respaldos almacenados (esto conlleva los respaldos, ¿Si existen, o están también infectados?). Sin olvidar las bitácoras esto en el caso de que aun confiemos en ellas y que existan.

3.3. Plan de contingencia

El plan de contingencia que está dentro de nuestras Políticas de Seguridad de Cómputo (PSC), en caso de que no poder comunicarse los superiores, de antemano que existe una intrusión y no es un falso positivo. En caso contrario se menciona una serie de recomendaciones a revisar, en su totalidad será un plan básico.

1. Crear una copia completa del sistema ya sea por cintas o un `iso`[1], en un metodo confiable, puede ser cintas o cd-rom o dvd; todo ésto y la firma digital del administardor, para poder hacer validez en caso de un acuse.

2. Al iniciar nuestra búsqueda tenemos que hacer por escrito en papel y en forma electronica de un reporte de cada acto encontrado, también podemos utilizacizar utilerias como `script` que nos ayudan porque graban desde que realisamos el analisis hasta que nos salimos de sesión en un archivo. A su vez el archivo lo podemos firmar para autenticarlo y utilizarlo como evidencia. Con casos muy particulares, se considera un filme o pelicula de analisis.

3. Se debera determinar en que se va a confiar en un sistema caido y comprometido, o en un sistema reistalado. aunque no se garantiza nada que no este infectado desde ese momento. Por otra parte se debera hacer serie de pasos manuales para que, al menos, datos basicos de que fue violado el servidor o servicion en cuestión.

4. Busca señales de que tu sistema ha sido comprometido

   Examinar las bitacoras del sistema, esto implica, conexiones de lugares inusuales u otra actividad inusual. Por ejemplo, busca tu último acceso al sistema, conteo de procesos, o todos los accesos generados por syslog y otros accesos de seguridad. Si tu firewa o

[1] archivo imagen para grabado en disco compacto o en una partición de un disco duro

ruteador escribe accesos a una diferente localidad que la del sistema comprometido, se debera de revisar estos accesos también. Nota que esto no es infalible a menos que tu acceses a un medio en el que solo sepueda añadir, muchos intrusos editan archivos log en un esfuerzo por esconder su actividad.

5. Busca archivos setuid y setgid (especialmente archivos setuid root) en todas partes de tu sistema.

   Tenemos que hacer una búsqueda de archivos, esta búsqueda debe de comprender, archivos modificados, archivos con últimas fechas de acceso o creación. Para realizar este trabajo nos valemos de las funciones de `ls, stat` y `find` con sus multiples extensiones, como por ejemplo:

   `ls -la *` esta instrucción nos ayudará con el último acceso; recordemos que debemos de ser root y estar en la raiz.

   `ls -lc *` con respecto a esta indicación, nos beneficiará para el ultimo cambio realizado.

   Y en caso de stat nos entrega la siguiente informacion, es similar a ls a diferencia que lo despliega en una sola pantalla y archivo por archivo.

```
[root@localhost root]# stat /etc/passwd
  File: "/etc/passwd"
  Size: 1226            Blocks: 8      IO Block: 4096 Regular File
Device: 303h/771d       Inode: 128607  Links: 1
Access: (0644/-rw-r--r--)              Uid: (0/root)  Gid: (0/root)
Access: Sun May 26 21:26:28 2002
Modify: Sun Jan 20 13:07:16 2002
Change: Sun Jan 20 13:07:16 2002
```

   Par poder entregar la confianza en los binarios que se ejecutan, debido que en algunos cosas, los rootkit modifican una serie de binarios, para esto se deberde revisar por ejmplo si se tiene tecnología rpm, o comparando los archivos del día cero cuando se instaló el servidor, o si el servidor puede entregarnos la firma del paquete esto se hace con md5 o sus variantes, y tambien el programa de tripwire.

   Al realizar una comparacion del medio compacto con el instalado en:

   `diff /bin/ps /mnt/cdrom/bin/ps`

   En este caso la salida no debe de entregarnos nada. En caso contrario, en caso de rpm seria:

   `rpm -V nombre_del_o_los_paquete(s)_comprometido(s)`

En su defecto hacer la revisión con md5 y su firma, la firma está concentrada en el caso de herramientas GNU en los sitios de distribución, o por su creador SUN, IBM, SISCO, 3com, etc.

Como administradores, o herederos de ellos debemos pedir archivos del dáa cero. Esto con el fin de hacer una comparación del sistema pasado y presente.

Los intrusos frecuentemente dejan copias shost, rhost o setuid de /bin/sh o /bin/time para así autorizarles el acceso como root a una ocasión posterior. El comando find puede ser usado para buscar este tipo de archivos. Por ejemplo, puedes usar los siguientes comandos para encontrar los archivos setuid root y los archivos setgid kmem en el completo sistema de archivos:

```
find / -user root -perm -4000 -print
find / -group kmem -perm -2000 -print
```

Lo anteriores ejemplos buscan en el directorio completo incluyendo NFS/AFS montados en el sistema de archivos. Algunos comandos find soportan la opción xdev"para evitar busccar esas jerarquías. Por ejemplo:

```
find / -user root -perm -4000 -print -xdev
```

Otro modo de buscar archivos setuid es usar el comando ncheck en cada partición de disco. Por ejemplo, usa el siguiente comando para buscar archivos setuid y equipos especiales en la partición de disco /dev/rsd0g:

```
ncheck -s /dev/rsd0g
```

6. Checa los archivos binarios del sistema para asegurarte que ellos no han sido alterados.

   He observado que algunos intrusos cambian programas en sistemas UNIX tales como login, su, telnet, netstat, ifconfig, ls, find, du, df, libc, sync, y archivos binarios referentes a /etc/inetd.conf, y otros programas críticos de sistema y la red y librerías de objetos compartidas. Compra las versiones en tus sistemas copias bien conocidas, tales como aquellas de su instalación inicial. Se cuidadoso al confiar en los respaldos, tus respaldos podrán contener tambien Troyanos.

   Los programas troyanos pueden producir el mismo checksum y timestamp estándar como la versión legítima. Debido a esto, el comando estándar de UNIX y los timestamps asociados con los programas no son suficientes para determinar si los programas han sido reemplazados. El uso de herramientas checksum como cmp, MD5, tripwire y otras herramientas checksum criptográficas son suficientes para detectar estos programas troyanos, provistas las herramientas checksum ellas mismas son mantenidas seguras y no están disponibles para modificaión por el intruso. dicionalmente, puedes querer considerar usar una herramienta (PGP por ejemplo) para "firmar"la salida generada por MD5 o Tripwire, para futura referencia.

7. Al revisar en tus sistemas uso no autorizado del progrma de monitoreo de red, comunmente llamado como sniffer o paquete sniffer. Los intrusos pueden usar un sniffer para capturar información de la cuenta y password de un usuario. Para localizar este tipo de casos, se debera de revisar los procesos si alguna nueva aplicaion esta generando un archivo o s comunica alguna parte de la maquina victima a cualquier parte del mundo. Esta solucion puede ser colocar la maquina atacanda dentro de un firewall para saber si se comunica al exterior o en el caso de la autogeneración de una bitacora de sniffer.

8. Examina todos los archivos que están correindo como los archivos "cron" y "at". Se ha visto que los intrusos dejan back doors en archivos corriendo como "cron" o enviados como "at". Estas técnicas pueden dejar un back intruso en el sistema. Además verifica que todos los archivos/programas relacionados (directa/indirectamente) por tareas del 'cron' y 'at', y las tareas se archiven por si mismas no sean escribibles.

9. Analisar si hay servicios no autorizados. Inspecciona /etc/inetd.conf por si tiene añadiduras no autorizadas o cambios. En particular busca entradas que ejecuten un programa shell (por ejemplo /bin/sh o /bin/csh) y cheaca todos los programas que esten especificados en /etc/inetd.conf para verificar que están correctos y queno han sido reemplazados porprogramas troyanos.

   Además hay que checar la legitimidad de los servicios que hayas comentado en el archivo /etc/inetd.conf. Los intrusos pueden habilitar un servicio que pienses que previamente lo habias deshabilitado, o reemplazan el programa inted con un programa troyano.

10. Examina el archivo /etc/passwd en el sistema y checa las modificaciones a ese archivo. En particular busca la creación no autorizada de nuevas cuentas, cuentas sin password o cambios de UID (específicamente UID 0) a cuentas existentes.

11. Cotejando las entradas no autorizadas a los archivos de configuración de tu sistema y de tu red. En particular busca ls entradas con signo '+' y nombres de host no locales inapropiados en /etc/hosts.equiv, /etc/hosts.lpd, y en todos los archivos .rhosts (especialmente root, uucp, ftp, y otras cuentas de sistema) en el sistema. Estos archivos no deberán de ser escribibles para todo el mundo. Por lo tanto confirma que estos archivos existan antes a cualquier intrusión y que no fueron creados por un intruso.

12. Realizar búsquedas de archivos o carpetas inusuales, estos archivos o carpetas vienen precedidos por lo menos de un punto, y utilizan una serie carpetas como ...(punto, punto, punto) o .. (punto, punto) hasta la utilización de caracteres ascii, pero como saber como se llaman debemos de buscar los tiempos y nombres de procesos. Un patron es de dejarlos en `/proc` o en `/usr/bin`. O cuando fue usada una cuenta de un usuario deja los archivos o en su `home` o en `tmp` o donde tenga permisos. Y a su vez, encontrar los archivos de `rootkit` y `backdoors`.

13. Busca en todas partes del sistema archivos ocultos (archivos que comienzan con un punto y normalmente no son mostrados por el "ls"'), como estos pueden ser usados para esconder herramientas e información (programas de crackeo de password, archivos de password de otros sistemas, etc). Una técnica comun en sistemas UNIX es poner un directorio oculto en una cuenta de un usuario con un nombre inusual, algo como "..." o ".." o "..Ĝ " . De nuevo el comando find puede ser usado para buscar archivos ocultos, por ejemplo:

```
find / -name ".. " -print -xdev
find / -name ".*" -print -xdev | cat -v
```

Además, archivos con nombres como '.xx' y '.mail' han sido usados (con esto se indica que son archivos que pueden parecer normales).

O al realizar una buesqueda por tiempos, en caso de que el ataque fue realizado hace 2 dias se debera realizar lo siguiente `find / -atime X`, donde X va ha tomar el valor de numero de dias que se pudo haber presentado el insidente, como atime, es la funcion de tiempo de acceso. Esto es va ha buscar desde raiz el ultimo acceso de X tiempo. Esto entre otras opciones que nos pueden beneficiar la función find.

14. Examina todas las máquinas en la red local cuando busques signos de intrusión. La mayoria de las veces, si un host ha sido comprometido, otros en la red lo han sido también. Esto es especialmente cierto para redes donde esta corriendo aplicaciones(NIS, NFS o clustering) o donde los hosts confáan uno en otro atravás del uso de archivos .rhosts y/o archivos /etc/hosts.equiv. Además, recomiendo los hosts para lo cuales tus usurios comparten acceso `.rhosts`.

15. Verificar consistencia esto es la revision de procesos corriendo fecha de solicitud, y quien lo esta generando el archivo.

En este caso debemos de conocer bien el sistema, como que es lo que esta brindando, debemos de ser meticulosos y desconfiar de todo proceso, porque existen procesos malignos que toman por nombre procesos de servicios comunes o hasta del kernell. En este punto es muy critico porque podemos encontrar que el atacante pude estar recibiendoi informacián o actuando el paquete instalado. Esto va ha depender si tomamos previamente la decisián de desconectarla de red o no.

Poner ejemplo:

Otra utilería que deben de contemplar en su efectividad es lsof, no todos los sistemas por default lo tienen, pero hace la búsqueda de archivos ejecutandose, conjuntamente a ps podemos hallar más rápido nuestro objetivo.

Básqueda de archivos ocultos como con permisos de SETUID y SETGID, con find podemos hacer la básqueda como por ejemplo

`find / -perm --2000 --print`, donde find va a buscar archivos con permiso 2000 y los va a imprimir en pantalla

`find / -user root -perm --4000 --print`, es similar pero hace la búsqueda específicamente con un usuario en particular, en el caso que se desconfiá de que alguien haya usado a root. Imaginemos en los casos de Irix 6.5 donde tiene una bitácora personalizada su, donde su es un comando de cambio de usuario, utilizado cománmente para cambiar a super-usuario, llamada sulog en la cual contiene cada acto que proceso desde el inicio de la instalación. En el caso de que no encontremos dicho archivo o un cierto tiempo, puede provenir de varias alternativas por ejemplo syslog.

16. La configuración de syslog, despuás de un tiempo lo manda a otro archivo que alguien lo haya destruido parcial o totalmente, así de esta manera podemos desconfiar en un usuario o en un grupo y la búsqueda por grupo la única modificación seria envás de `--user` seria `--group` y el nombre del grupo. En los casos de los servidores de impresión y el Worm de Ramen se apoderaba de lp, así que se hacia la búsqueda del usuario lp o del group de root.

17. Eliminacián de rootkit:

18. Eliminacián de rootkits

    Para realizar esta accion se debe de tomar en cuenta de la condicion más optima es desconectada de la red, y aislada hasta determinar el impacto de la infeccion y para restringir a posibles amenazas. Para esto, se necesitaria el apollo de la empresa u organizacion para un servidor mirror instalado para que continue con las tareas del servidor victima.

    Es necesario buscar en los comandos: `df du, ls, ps-[fea|aux], netstat, lsof`, si nos asepta esta instruccián "-/". Esto sumado al gran consumo de procesador y de disco duro, pensando que instalaron un sniffer o sobrecarga en la red. Como tambien algunos comandos hacen cosas raras, el disco duro se satura, a red esta lenta, ejecucion de procesos desconocidos, coneciones de maquyinas desconocidas, usuarios nuevos o dumplicados de nombre pero no de ID, o se intentan conectar sin exito a hora inusuales, reporto barrido de puertos, caidas desistema, sesiones imposibles, reportes de usuarios o administradores, hasta el porque de los KERNELL PANIC o reinicios de servidor.

    Otra manera es comparar los binarios de origen del sistema con los del posible ataque. En los casos de no tener nada valioso en el sistema, lo más sano es reinstalar, desafortunadamente al reistalar vamos a hacer el mismo error del bug con el que ingresarón. Y por el otro lado, si se tiene información, que posiblidad exista de que este en nuestro respaldo.

Para hacer un análisis necesitaremos de herramientas que no necesariamente son de licensia:

para detectar sniffer con netstat, ifstatus, nePED a su vez los caballos de troya los encontraremos con sum, cmp, md5, tripwire y COPS con el modulo de crc.chk de borrado de rastros, por ejemplo antizap.c y análisis de red con Netcat, AAFID, ISS,Nessus

Por ejmplo swatch, el cual es un analisador de bitacoras, en el podemos filtrar mucha informacion, esto solo configurando y enrrutando muchas acciones de administracón.

Logwatch creado por psionic el cual manda cualquier accion relevante al encargado, solo se configura a nuestras necesidades y todo nos lo manda por correo.

3.4. Herramientas para el Analisis Forense

Para optimizar el analisis forense se utilizo Software de libre distribución. A continuación se detallan algunas de las herramientas utilizadas:

bridge-utils: Algunos de los comandos que incorpora son:

- addbr añade un puente
- addif añade una interfaz a un puente
- delbr borra un puente
- delif borra una interfaz de un puente
- show muestra una lista de puentes
- showbr muestra informacián sobre un puente
- showmacs muestra una lista de direccionés MAC[2]

tcpdump: Básicamente, tcpdump es una herramienta a nivel de trasporte que imprime los encabesados de los paquetes que pasan por una interfaz de red determinada y que se corresponden con un determinado patrón prefijado. Tcpdump incorpora multitud de opciones que permiten especificar patrones de comportamiento de los paquetes que se desea capturar.

Por las características del programa, puede ser usado para supervisar y la detección de intrusiones en sistemas informáticos.

[2]MAC: dirección maquina obtenida de la tarjeta de red.

snort: Es un sniffer/logger de paquetes flexible para detectar ataques. Snort esta basado en la biblioteca libpcap que puede ser utilizadas como sniffer/logger de sistema de detección de intrusos en redes de poca carga. Las caracteristicas del logging estan basadas en reglas que pueden ser representadas por busqueda/ concordancia del contenido, adicionalmente puede ser utilizado para detectar gran variedad de otros ataques y pruebas, como los buffer overflow, escaneo de puertos sigilosos, ataques CGI, SMB, y mucho mas. Snort tiene la capacidad de alerta en tiempo real, con alertas enviadas a syslog a un archivo separada de "alerta", o incluso a una computadora Windows a traves de Samba. Snort es el mejor sniffer/logger que hay en estos momentos para pequeñas redes, el uso de reglas sencillas y el uso de fingerprint hace sencillo el detectar cualquier ataque que atente contra nuestro sistema. Algunas de los detalles que se le puede echar en cara en mi opinion es la imposibilidad de loggear la direccion ip del atacante, aunque esto se puede solventar utilizando conjuntamente con tcpdump. La principal caracterástica que diferencia a snort de tcpdump es que el primero puede inspeccionar la carga de datos (payload) de los paquetes.

iptables: Es un firewall. Tal herramienta iptables se utiliza para configurar, mantener e inspeccionar las tablas de reglas de filtrado de paquetes IP del núcleo de un sistema Linux. Iptables permite definir varias tablas distintas. Cada tabla contiene un cierto número de cadenas predefinidas, como añadir otras cadenas definidas por el propio usuario.

Cada cadena es una lista de reglas que pueden corresponder con un conjunto de paquetes.

Cada regla especifica qué se debe hacer cuando un paquete coincide con el patrón que contiene la propia regla.

Iptables resulta muy útil cuando se desean establecer reglas que permitan realizar el filtrado de equipos atacados (es decir, cortar la conexión a esos equipos) para su posterior análisis.

dd: Esta herramienta sirve para convertir y realizar copias a nivel bit de archivos o particiones completas a nivel de bloques de datos. Viene, prácticamente, con todas las versiones de Unix. Es de gran utilidad para realizar análisis forense, ya que por sus caracterásticas es fácil de usar y no modifica los tiempos de acceso a los archivos.

nc (NetCat): Netcat (nc) es una utilidad para equipos Unix que permite leer y escribir datos a través de conexiones de red, usando el protocolo TCP o UDP. Netcat se diseñó para que pudiera ser usado fácilmente tanto de manera directa como integrado en otros programas o en scripts.

Debido a sus características, es ideal para realizar el volcado de las imágenes de las particiones de los equipos atacados sobre el sistema donde se procederá al análisis de

las mismas.

script: El comando script hace una trascripción de cualquier cosa que sea impresa en un terminal. Es útil cuando se están copiando los datos después de una intrusión, para que queden grabadas todas las acciones realizadas.

The Coroners Toolkit (TCT): Es un paquete forense de libre distribución que puede ayudar en el análisis de imágenes de particiones del tipo FFS (propias de sistemas BSD y Solaris) y del tipo EXT2FS (propias de Linux). El paquete incluye una potente biblioteca en C, fs_lib.a, y varias herramientas que pueden ser usadas sobre particiones montadas o bien sobre imágenes de estas particiones. A continuación se describirón algunas de las herramientas que componen el paquete:

grave-robber y mactime: La herramienta grave-robber es un programa encargado de recoger datos analizando imágenes de sistemas de archivos, dando como resultado, estar el sistema activo (en tiempo real) o fuera de uso (post-mortem).

Cualquier archivo de un sistema de archivos *FFS* o de un *EXT2FS* almacena tres tiempos: Tiempo de Modificación, Tiempo de Acceso y Tiempo de Cambio (conocidos como tiempos *MAC*). Cuando se pasa la opción '-m', grave-robber guarda los tiempos *MAC* de cada archivo y de cada directorio en un archivo llamado "body". La herramienta también puede guardar otros datos como son los archivos borrados que todavía están abiertos y contenidos en la memoria del sistema. Por otro lado, la utilidad `mactime` es un script en perl que procesa el contenido del archivo "body" (generado por grave-robber). Se encarga de crear un archivo ASCII con una línea temporal, donde cada entrada corresponde a una modificación, acceso o cambio de uno de los archivos. Una línea temporal resulta útil cuando se está intentando determinar qué archivos han sido creados o accedidos recientemente. Más aún, debido a que muchos sistemas tienen gran cantidad de espacio, la línea temporal puede mostrar los directorios donde el investigador debería centrar su foco de atención. Por supuesto, un atacante puede modificar fácilmente los tiempos *MAC* y si pasa mucho tiempo, los datos relacionados con el atacante pueden ser sobrescritos.

ils y ils2mac: La herramienta ils muestra datos sobre los i-nodos no asignados (unallocated). Entre la información que recoge está el tamaño del archivo y los tiempos MAC. El script ils2mac convierte la salida de ils al mismo formato que tiene el archivo "body" generado por grave-robber. Por tanto, se puede concatenar la salida de ils2mac dentro del archivo "body" y generar una línea temporal con entradas tanto de los i-nodos asignados como de los no asignados. Esto proporcionará pistas sobre cuándo y dónde estaban los archivos borrados.

unrm y lazarus La herramienta unrm es una variante de dd y produce un flujo de bloques de contenido. Por defecto, extrae los bloques no asignados de la imágenes de la

partición. En otras palabras de un archivo borrado se puede reconstruir por bloques.

El objetivo de lazarus es "crear estructuras a partir de datos no estructurados". Lazarus toma un flujo de bytes como entrada y los analiza en trozos del tamaño de un bloque. Es decir, trata de identificar qué tipo de datos contiene el bloque (por ejemplo, código C, un archivo tar, correo) y crea un archivo que contiene una lista con el tipo predicho de cada bloque y un directorio de archivos que contienen los bloques. Opcionalmente, la salida de lazarus puede ser generada en HTML y con un navegador se puede examinar el contenido de cada bloque. Cuando lazarus se usa sin unrm, los bloques no asignados puede ser seleccionados para buscar el contenido borrado.

Este proceso puede ser tedioso en particiones grandes con gran cantidad de espacio libre.

icat: La utilidad icat muestra el contenido de un archivo o un directorio, que es especificado mediante un i-nodo y la imagen donde se encuentra. Es similar al comando cat de UNIX, con la diferencia de que en vez de usar el nombre del archivo como argumento se usa el i-nodo. Por tanto, icat puede ser usado para ver el contenido de i-nodos no asignados (siempre que el sistema no haya borrado los punteros de bloque).

TCTUTILs: Aunque TCT proporciona herramientas muy potentes para el análisis forense, le falta, funcionalidad. Por ejemplo, la posibilidad de listar los nombres de los archivos y de los directorios que hay en una imagen, la posibilidad de mapear los bloques y los nodos-i, la posibilidad de mapear los nombres de los archivos con los nodos-i, y la posibilidad de obtener detalles de un i-nodo específico. Por esa razón fue desarrollado TCTUTILs. TCTUTILs proporciona herramientas adicionales que vienen a suplir esta falta de funcionalidad. Algunas de las herramientas que están incluidas en TCTUTILs son.

istat: Istat muestra todos los datos conocidos sobre un i-nodo, incluyendo todos los bloques que tiene listados en sus punteros a bloque, y las direcciones de los bloques indirectos. Esto es útil para la generacián de informes sobre los nodos-i y para proporcionar un formato más nítido que el que se obtiene con la herramienta ils de TCT.

bcat: Esta herramienta permite desplegar el contenido de un bloque. Su contenido puede ser mostrado como cádigo ASCII, hexadecimal o formato binario. También es posible mostrar la salida como una tabla HTML.

find_inode: Esta utilidad busca la imagen de un i-nodo que tenga, en su lista de apuntadores a bloques, un bloque dado. Se pueden dar tres posibles situaciones cuando es utilizada: el bloque está en la lista del i-nodo, el bloque no está en la lista del i-nodo, o el bloque está contenido dentro de un fragmento más amplio.

fls: Lista los archivos y los directorios que tienen entradas en un bloque de directorio asignado (allocated). Esta herramienta tiene muchos usos obvios. Primero, puede listar todos los archivos borrados para lograr una mejor comprensión de qué herramientas ha instalado el intruso en el sistema. El segundo uso es permitir ver el contenido de la partición sin tener que montarla mediante el mecanismo de loop back, cosa que es útil cuando no se dispone de esa opción.

Otra posibilidad que da, es imprimir datos en el formato MAC. Esto permitirá obtener estadísticas de los archivos borrados del sistema cuyo valor de i-nodo no haya sido borrado en la entrada del directorio. Como el formato de salida es compatible con el generado por grave-robber (TCT) se pueden combinar ambos archivos y procesarse posteriormente por mactime (TCT).

find_file: Esta herramienta se mete recursivamente en los directorios, empezando por el directorio raíz, y busca una entrada que tenga como i-nodo el pasado como parámetro.

blockcalc: Realiza la conversión entre los números de bloque de una imagen generada con *unrm* y los números de bloque de la imagen original. Una imagen generada con *unrm* contiene un subconjunto del total de bloques que hay en la imagen original y, por tanto, no hay un mapeo obvio con los originales. *Blockcalc* crea un mapa entre las imágenes, y puede convertir el número de bloque de una imagen en el número de bloque de la otra.

Ethereal: Ethereal es una herramienta de libre distribución que actúa como analizador de protocolos de red, y puede ser utilizado tanto en equipos Unix como en equipos Windows. Permite examinar los datos procedentes de una red activa o bien almacenados en un archivo (obtenidos, por ejemplo, de la ejecución de tcpdump).

3.5. Caso practico

A continuación se describen los resultados obtenidos, las líneas de investigación seguidas y los principales objetivos alcanzados.

Una forma de monitorear sin comprometer excesivamente la integridad de un sistema es mediante el proceso denominado "jailing" (más conocido como honeypot) o encarcelamiento: la idea es construir un sistema que simule a uno real, pero donde no se encuentren datos importantes, y que permita observar al atacante sin poner en peligro los sistemas reales. Para ello se utiliza una máquina, denominada sistema de sacrificio, máquina trampa o equipo víctima, que es donde el atacante realmente trabaja, y un segundo sistema, denominado de

observación, de control o de monitorización, conectado al anterior y que permite analizar todo lo que esa persona está llevando a cabo. De esta forma se logra que el atacante piense que su intrusión ha tenido éxito y continúe con ella mientras lo monitorizamos y recopilamos pruebas para presentar en una posible demanda o acusación. En lo que sigue a continuación se muestran los pasos dados para la implementación de este sistema y los resultados obtenidos del análisis forense de los equipos atacados.

3.5.1. Implementación de la arquitectura

Uno de los objetivos de este proyecto, como ya se ha comentado anteriormente, es montar una infraestructura de equipos informáticos que permita, por un lado, monitorear los ataques que se produzcan sobre estos sistemas y, por otro, analizar y almacenar toda la información de las conexiones que se realicen sobre los equipos para su procesamiento posterior.

Topología

Cuando se plantea la elección de una topología para una arquitectura, se deben observar 3 objetivos principales:

Proporcionar máxima fiabilidad que garantice la correcta recepción de todo el tráfico.

Encaminar el tráfico entre transmisor y receptor a través del camino más económico y confiable.

Proporcionar un tiempo de respuesta óptimo.

En este caso, las últimas dos características no son excesivamente importantes y vienen garantizadas por la propia infraestructura preexistente sobre la que monta nuestra arquitectura.

Es evidente que para lograr detectar cualquier intento de ataque contra la red de equipos víctima, necesariamente toda la información destinada a éstos debe pasar, previamente, por el equipo de control. Una consecuencia inmediata de esto es que es necesario que estos equipos están, en cierta forma, aislados, de manera que toda comunicación con el exterior se centralice a través del sistema de control.

También es inmediato pensar que se debe arbitrar algún mecanismo que permita comunicar al equipo de control con los equipos trampa sin necesidad de que haya una conexión física directa entre ellos, puesto que el número de interfaces de red que debería tener el equipo de control tendría que ser igual al número de equipos a monitorear (una más si tenemos en cuenta que también debe tener salida a Internet).

Usar un HUB permite meter tráfico falso dentro de la red que simule conexiones de usuarios ficticios (conexiones telnet, pop3, ftp, etc.) de forma que si el atacante dispone de un sniffer podrá capturar logins y passwords, permitiendo observar y analizar el patrón de comportamiento del atacante: ver si intenta conectarse a los equipos mediante la información obtenida, por ejemplo.

Además, en caso de saturación de la red (Internet), la conexión entre la máquina de captura y los equipos trampa no se bloquea en ningún un momento. Por otro lado, otra ventaja del HUB es que, como máximo, permite la salida de tráfico a 10 Mb, mientras que la interfaz de salida al exterior del sistema de control funciona a 100 Mb, por lo que tampoco supondrá un problema.

Resumiendo, lo visto anteriormente, la mejor configuración para nuestro sistema seráa una topología en estrella, ya que es fácil de controlar, no necesita software adicional y el flujo de tráfico es sencillo. Además, todo el flujo de información pasa por el equipo de control, lo que permite encaminar y monitorear todo el tráfico que llega a los equipos trampa, localizar disfunciones de éstos y aislar individualmente a cualquiera de ellos (mediante el establecimiento de reglas de filtrado de paquetes basadas en la dirección IP de la máquina de destino o en su dirección MAC).

3.5.2. Preparación del equipo de control

Una vez diseñada la topologáa sobre la que se va a situar la arquitectura se procederá a la configuración del equipo de control. Básicamente, se trata de montar un IDS (Intruder Detection System o Sistema de Detección de Intrusos) y poner en marcha de mecanismo que permita la automatizacián del proceso de detección de ataques, la rotación y actualización de logs, y el filtrado de equipos, cuando sea necesario. Los pasos seguidos en este proceso son los que detallan a continuacián:

Instalación del sistema operativo

El sistema operativo utilizado en el equipo de control es OpenBSD 3.2, una versión de Linux en la que se refuerzan los aspectos de seguridad, e instalación mínima. Además, con el objeto de que el sistema operativo fuera lo más estable y confiable posible, se actualizó con la última versión del núcleo disponible en ese momento.

Los motivos de la elección de Linux como sistema operativo son evidentes: como todo sistema Unix es potencialmente seguro, es un aspecto de gran importancia en el proyecto, puesto que, debido a las características del equipo, un fallo en la seguridad podría acabar con muchas horas de trabajo. Además, Linux tiene la ventaja de soportar sistemas de archivos de varios sistemas operativo distintos (como, por ejemplo, Sun UFS), lo que facilita la labor de análisis forense. Otra ventaja de usar Linux son sus dispositivos de loopback, que permiten montar un archivo con la imagen de una partición (obtenida con la herramienta dd) de un equipo atacado, dentro del propio sistema de archivos de Linux.

Funcionamiento en modo bridge

Si se desea obtener información sobre el tráfico que circula por la red con destino a los equipos víctimas analizando los paquetes que circulan por ella de modo transparente, es necesario configurar el sistema para que realice bridging.

Para lograr este objetivo se puede hacer uso de la herramienta bridge-utils. Una vez instalada en el equipo de control, se deberá comprobar que las tarjetas de red funcionan correctamente y están accesibles. Para eso, el comando ifconfig puede ayudar mostrando información sobre el estado de éstas.

Así mismo, el núcleo del sistema operativo debe tener activada la opción que permite hacer bridging: `CONFIG_BRIDGE=Y`

Después de estos primeros pasos, si se ejecuta el comando

```
modprobe ?v bridge
```

y, si todo lo realizado hasta ahora ha sido correcto, no debería mostrar ningún error.

A continuación se puede proceder con la configuración básica de puente, que consistiría en:

1. Crear la interfaz del puente. `# brctl abr mipuente`
2. Añadir interfaces al puente. `# brctl addif mipuente eth0 # brctl addif mipuente eth1`
3. Poner a cero las direcciones IP de las interfaces. `# ifconfig eth0 0.0.0.0 # ifconfig eth1 0.0.0.0`
4. Activar el puente. `# ifconfig mipuente up`
5. Opcionalmente, se puede configurar la interfaz virtual de puente para que forme parte de la propia red (sin que por ello el puente deje de ser transparente). De esta forma se consigue que se comporte como otra interfaz más (es decir, como una tarjeta de red normal). Para ello bastará con sustituir el paso anterior por un comando como este:

   `# ifconfig mipuente 155.54.xxx.xxx netmask 255.255.xxx.xxx up`

   (donde xxx representa un valor válido para la dirección IP) Para que el puente se active cada vez que se reinicie el equipo, se deberá crear, además, un script con los pasos anteriores. Este script se lanzará en tiempo de arranque.

Activación de las iptables.

Como señalo anteriormente, es necesario disponer de un mecanismo que permita bloquear el acceso a los equipos trampa desde el equipo de control. Esto se consigue mediante la utilización de las iptables. En Linux (el equipo de control utiliza este sistema operativo), el filtrado de paquetes está construido en el kernel (núcleo); por tanto, para poder utilizar iptables (es decir, para que el equipo de control pueda actuar como firewall) se ha de compilar éste con las opciones adecuadas para que la infraestructura de filtrado de red está activa. Básicamente se reduce a activar las opciones:

```
CONFIG\_FIREWALL
CONFIG\_IP\_FIREWALL
CONFIG\_IP\_NF\_IPTABLES
```

Una vez que el núcleo está ejecutándose con el firewalling activado, se utilizará iptables para insertar y eliminar reglas de filtrado en él; al tratarse de información dinámica, cada vez que el sistema reinicie las reglas establecidas se perderán, por lo que es recomendable crear un script que se ejecute al arrancar el sistema y que las vuelva a definir (siempre que, claro, se quieran hacer permanentes).

El núcleo de Linux utiliza tres listas de reglas denominadas chains o cadenas, se trata de input, output y forward. Cuando recibe un paquete utiliza la primera de estas listas para decidir si lo acepta, y si es así comprueba a dónde tiene que enrutar el paquete; en caso de que el destino sea una máquina diferente utiliza las lista forward para enviarlo. Por último, la lista output se utiliza, obviamente, antes de enviar un paquete por un interfaz de red.

Los elementos de cada lista se denominan reglas y definen qué hacer con los paquetes que cumplen ciertas características. Si un paquete no cumple ninguna de las reglas que deciden qué hacer con él , se aplica la política por defecto: en estos casos lo más seguro es rechazarlo.

Cuando un paquete cumple una determinada regla de una chain definimos qué hacer con él mediante lo que iptables denomina objetivo o target. Aunque existen más targets, son dos los que más se suelen utilizar: ACCEPT permite el paso de un paquete y DENY lo bloquea.

Para mostrar el funcionamiento real de la herramienta, se supondrá, por ejemplo, que se desea bloquear todo el tráfico que pasa por la máquina de control con destino u origen a uno de los equipos víctima, cuya dirección IP está dada por DIR_IP_VICTIMA.

La secuencia de comandos que se tendría que ejecutar seráa las siguiente:

```
# iptables ?A INPUT -s DIR_IP_VICTIMA -j DENY

# iptables ?A INPUT ?d DIR_IP_VICTIMA -j DENY

# iptables ?A FORWARD -s DIR_IP_VICTIMA -j DENY

# iptables ?A FORWARD ?d DIR_IP_VICTIMA -j DENY

# iptables ?A OUTPUT -s DIR_IP_VICTIMA -j DENY
```

Si en vez de bloquear usando la dirección IP del equipo se quiera usar la dirección MAC del mismo, se tendráa que hacer lo siguiente:

```
# iptables ?A INPUT ?m mac --mac-source DIR_MAC_VICTIMA -j DENY

# iptables ?A FORWARD -m mac -- mac-source DIR_MAC_VICTIMA -j DENY
```

En ambos casos, con la opción A se está indicando que se añade la regla a la chain especificada (INPUT, OUTPUT, FORWARD); "-s" permite especificar la dirección de la máquina origen, "-d" permite especificar la dirección de la máquina destino y "-j" indica el objetivo, en este

caso DENY. La opción "m mac" activa la utilización de las direcciones MAC, y "– mac-source" permite especificar la dirección MAC del equipo origen.

Funcionamiento en modo bridge+firewall

En los dos apartados precedentes se han mostrado los pasos necesarios para configurar el equipo de control de forma que pueda realizar bridging y firewalling; en este apartado se pretende dar una visión de cómo combinar ambos elementos. Uno de los problemas que surgieron al intentar usar conjuntamente el bridge y las reglas de filtrado de paquetes fue la incompatibilidad que presentaban estas últimas en su modo de funcionamiento con la activación del bridge. Para solucionar este problema se dieron los siguientes pasos:

1. Aplicación de un parche para compatibilizar el funcionamiento del bridge con el filtrado de paquetes.
2. Compilación del núcleo con las opciones necesarias para realizar filtrado de paquetes y funcionar en modo bridge.
3. Activación del módulo br_passthrough.o, que se encuentra en

```
/lib/modules/2.4.6/kernel/net/brdge/netfilter:
insmod /lib/modules/2.4.6/kernel/net/brdge/netfilter/br\_passthrough.o}
```

Monitoréo de los equipos

La revision del tráfico que circula por la red es indispensable para la seguridad de cualquier sistema: ésto nos facilitará información sobre los intentos de ataques que puede suceder (origen, franjas horarias, tipos de acceso...), así como la existencia de tramas que aunque no suponga un ataque apriori sí que son al menos sospechosas (por ejemplo, un escaneo de puertos).

Para detectar ataques (o intentos de éstos) se emplean dos vías. Por un lado se analiza en tiempo real el tráfico de red destinado a los equipos trampa '?utilizando para ello herramientas como tcpdump o snort?, y por otro se almacena toda la información relacionada con las distintas conexiones que se van produciendo, con el objetivo de procesarlas posteriormente en busca de información relevante.

Existen dos enfoques posibles para la detección de ataques a partir de la información obtenida:

- Definir los patrones de comportamiento de los ataques, de modo que cuando se detecte uno, avise al administrador (mediante un correo, por ejemplo).
- Definir los patrones normales, avisando cuando se detecte uno que no se ajuste a los estándares predefinidos.

El primer enfoque es incapaz de detectar ataques que no se hayan previsto (y es prácticamente imposible prever todos los ataques, sobre todo si se tiene en cuenta que cada día surgen nuevos métodos de intrusión). Sin embargo, eta interfaz tiene la ventaja de que no dará falsas alarmas y, si el conjunto de reglas que definen los patrones de los ataques es suficientemente completo, la mayoría de los métodos de ataque que circulan por Internet serán detectados.

Para tal efecto, snort es una herramienta muy completa, ya que permite definir reglas para la detección de ataques y asociar a éstas la acción que se considere oportuna, además de proporcionar y actualizar una suite de patrones que contemplan la mayoráa de las vulnerabilidades que existen en la actualidad.

El segundo de los enfoques detectará cualquier situación anómala, por lo que normalmente recogerá más ataques que el primero pero, por el contrario, capturará muchas falsas alarmas.

Un ejemplo de utilización de snort podría ser el que se muestra a continuación.

```
snort ?Dds ?c /etc/snort/snort.conf ?i eth1 ether host DIRMAC
```

La opción ??D? permite la ejecución de snort en segundo plano (es decir, como demonio), ?-d? descarga el tráfico del nivel de aplicación,?-s? envía mensajes de alerta al syslog, ?-c? permite indicar el archivo de reglas que se va utilizar para la captura, ?-i? especifica la interfaz por la que se escucha, y ?ether host? indica que únicamente se capturen los paquetes que provengan del equipo cuya dirección MAC coincida con DIRMAC.

Preparación de los equipos victima

Es la parte de instalación, para obtener mayor posiblidad de atacado se realiza una instalación por defecto de los distintos sistemas operativos. Apertura de todos los puertos y servicios disponibles. Creación de usuarios ficticios con logins y passwords fáciles de romper.

Instalación de herramientas que faciliten la migración de datos (necesario para cuando se quieran enviar los datos del equipo atacado a un equipo remoto para su posterior análisis).

Copia de seguridad de la información que contiene el sistema originalmente. Se puede usar para este fin una partición oculta del propio sistema. Sincronización de la hora del sistema con algún servidor de tiempo fiable: la idea es que tanto el equipo de control como los equipos trampa tengan (y mantengan) la hora sincronizada con algún punto de referencia común . Para lograr esto se puede utilizar el servidor NTP. La importancia de la sincronizacián reside en la posibilidad de comparar la información almacenada en el sistema de control con la que queda registrada en los equipos víctima (después de un ataque), sin necesidad de realizar conversiones horarias producto de desfases entre los relojes de los sistemas.

3.5.3. Análisis de ataques

Proporcionaré algunos patrones habituales de ataques y mostraré las actividades necesarias para realizar un análisis forense.

Ideas generales

La mayoría de las intrusiones terminan con que el atacante tenga permisos de root, suelen seguir el mismo patrón de comportamiento:

1. Primero el atacante realiza un escaneo buscando equipos vulnerables que están ejecutando un servidor con algún fallo de seguridad conocido y que se ha comentado, probablemente, en listas de seguridad, por ejemplo los fallos de desbordamiento de buffer en el servidor de FTP wuftp o del proceso rpc.statd.

2. El atacante emplea un exploit contra el equipo, consiguiendo instalar una puerta de acceso en el sistema. Muchas veces el exploit genera directamente un interprete de comandos con privilegios de root, o añade una línea o sentencia en el archivo /etc/inetd.conf para lanzar un shell en un puerto dado.

3. El atacante instala o compila un rootkit, conjunto de programas de nombre y comportamiento similar al de comandos del sistema operativo, que sin embargo no muestran información sobre determinados estados del sistema.

4. El atacante instalará y/o compilará algunas herramientas de ataque para escanear otros equipos y redes, empleando la máquina recién atacada como puente. Esta situación se prolonga hasta que alguien detecta un comportamiento anómalo en el equipo. Algunas veces esta detección se realiza por el propio administrador del equipo debido a una carga de procesamiento anormal, accesos extraños, etc., pero en la mayoría de los casos la detección del equipo atacado se produce desde el exterior: llega un aviso a la organización indicando que el equipo en cuestión está escaneando o ha sido empleado para atacar otros sistemas.

Estrategias de Respuesta

Existen dos estrategias de respuesta ante un incidente de seguridad:

Proteger y proceder: Sele aplicar cuando la organización es muy vulnerable o el nivel de los atacantes es elevado. La filosofía es proteger de manera inmediata la red y los sistemas, y restaurar su estado normal, de forma que los usuarios puedan seguir trabajando normalmente. Será necesario interferir de forma activa las acciones del intruso para evitar más accesos, y analizar el daño causado. La principal desventaja de esta estrategia es que el atacante se da cuenta rápidamente de que ha sido descubierto, y puede emprender acciones para no ser identificado, lo que incluso conduce al borrado de logs o el sistemas de archivos completos. Sin embargo, esta estrategia también presenta una parte positiva: el bajo nivel de conocimientos de los atacantes en sistemas habituales hace que en muchas ocasiones se limiten a abandonar su ataque y dedicarse a probar suerte con otros sistemas menos protegidos en otras organizaciones.

Perseguir y procesar: Se adopta la filosofáa de permitir al atacante proseguir sus actividades, pero de forma controlada y observada por los administradores. Con esto, se intentan guardar pruebas para ser utilizadas en la segunda parte de la estrategia, la de acusacián y procesamiento del atacante (ya sea ante la justicia o ante los responsables de la organizacián, si se trata de usuarios internos).

Evidentemente se corre el peligro de que el intruso descubra la monitorización y destruya completamente el sistema y que los resultados obtenidos no se tengan en cuenta ante un tribunal. La parte positiva de esta estrategia es aparte de la recolección de pruebas, que permite a los responsables conocer las actividades del atacante, qué vulnerabilidades ha aprovechado para atacarla, cómo se comporta una vez dentro, etc. De esta forma se puede aprovechar el ataque para reforzar los puntos débiles del sistema.

De las dos estrategias presentadas aquí, se utilizará la segunda, ya que es la que mejor se adapta los propósitos de este proyecto.

Principios para la recopilación de evidencias

El propósito de este apartado es facilitar unas directivas para la recolección de evidencias y el almacenamiento de pruebas después de producirse un incidente de seguridad que haya acabado con el acceso del atacante al sistema.

Aquí se describen los pasos a seguir si se decide recabar y proteger toda la información relativa a la intrusión. **Principios generales**

- Contactar con la autoridad competente encargada de tratar ese tipo de incidentes.
- Intentar capturar el estado actual del sistema como si se tratase de una foto.
- Guardar notas detalladas de todos los pasos dados. En éstas se deberá incluir horas y fechas en las que se realizá cada acción. Si es posible, generar una copia de manera automática (por ejemplo, en sistemas Unix, se puede usar el programa 'script', aunque el archivo de salida generado no deberá formar parte de las evidencias). Todas las evidencias y salidas obtenidas deberán ser firmadas y fechadas.
- Estar preparado para testificar (posiblemente años después), esquematizando todas la acciones realizadas y la hora a la que se llevaron a cabo. Obtener unas notas minuciosas es de vital importancia.
- Reducir al mínimo las alteraciones que se realicen sobre los datos durante el proceso de recolección. Esto no se limita únicamente a no modificar el contenido de los datos, sino que se deberá evitar modificar los tiempos de último acceso de los archivos y de los directorios.
- Intentar minimizar los accesos externos del investigador al sistema, así como determinar qué modificaciones son debidas al funcionamiento normal del equipo.
- Cuando se tenga que decidir entre realizar recolección de datos o realizar análisis, se deberá hacer primero la recolección y más tarde el análisis.
- Aunque difícilmente sea necesario demostrarlo, los procedimientos empleados deberán ser implementables. Como en cualquier incidente donde haya intervención policial, los procedimientos deben ser testados para asegurar su fiabilidad, especialmente en una

situación de emergencia. Si es posible, los procedimientos se deberán automatizar por razones de rapidez y de precisión. Se debe ser metódico.

- Cada dispositivo del sistema debe ser tratado de forma metódica, siguiendo las directivas determinadas en la guáa de recolección de evidencias. La velocidad muchas veces resulta crítica cuando hay varios dispositivos que requieren ser examinados al mismo tiempo, por lo que en estos casos puede resultar apropiado dividir el trabajo entre distintos miembros de la organización para paralelizar el proceso. Sin embargo, si se trata de un sistema con un único dispositivo, la recolección de datos se deberá realizar paso a paso.
- Proceder a la recolección de pruebas por orden decreciente de volatilidad
- Hacer una copia a nivel de bit de la información del sistema. Si se desea realizar análisis forense, se deberá hacer una copia a nivel de bit para este propósito, ya que cuando se proceda a realizar el análisis de la información, casi con toda certeza, se alterarón los tiempos de acceso a los archivos. Evitar hacer análisis forense sobre la copia destinada a servir de evidencia.

Orden de Volatilidad

Cuando se lleva a cabo la captura de evidencias, se deberá empezar por aquellas que tienen mayor probabilidad de desaparecer. La siguiente lista muestra un ejemplo del orden de volatilidad para un sistema tápico:

- Registros y caché.
- Tabla de enrutamiento, caché ARP, tabla de procesos, estadísticas del sistema.
- Archivos temporales del sistema.
- Disco.
- Logs y datos monitorizados de forma remota
- Configuración física, topográfica de la red

Pasos que se deberan evitar. En el proceso de recolección de pruebas es muy fácil eliminar evidencias inadvertidamente, por eso es importante que se observen las siguientes medidas:

- No apagar el sistema hasta que se haya completado la recolección de evidencias.

  Muchas evidencias se podrán perder y, además, el atacante puede haber modificado los scripts/servicios de arranque/parada de forma que eliminen pruebas de su presencia en el sistema.

- No confíe en los programas instalados en el sistema. Ejecute sus programas de recolección de pruebas desde medios seguros y externos al sistema.

- No ejecute programas que modifiquen los tiempos de acceso de los archivos del sistema (como, por ejemplo, 'tar' o 'gzip').

- Tenga en cuenta que cuando desactive alguna de las conexiones del sistema puede hacer ejecutar alguna aplicaión que detecte el analisis y elimine evidencias.

Consideraciones de Privacidad

- Respete las leyes de privacidad de su compañía y de la jurisdicción vigente. En particular, asegúrese de que ninguna de la información recolección con las evidencias sea información que normalmente no está disponible para cualquier persona. Esto incluye tanto accesos a los archivos de log (que pueden revelar patrones de comportamiento de los usuarios) como a los archivos de datos personales.

- No invada la privacidad de la gente sin una justificación sálida. Más concretamente, no recoja información de áreas donde normalmente no tiene razones para acceder, salvo que tenga indicios razonables de que realmente ha habido un incidente que afecta a la seguridad.

- Asegúrese de que su compañía respalda el procedimiento empleado durante la recolección de evidencias.

Consideraciones de Legales

La evidencias obtenidas del sistema deben ser:

- Admisibles: Deben de ser conformes con las leyes actuales para que puedan ser presentadas ante un jurado.

- Auténticas: Deben estar absolutamente accesibles.

- Completas: Deben mostrar todo lo que ha pasado en el sistema y no una perspectiva particular.

- Confiables: No debe haber ninguna duda sobre cómo fueron obtenidas.
- Creíbles: Deben ser realmente demostrables y comprensibles para los miembros de un jurado.

Todo esto puede ser valido con la asesoria de algun despacho de abogados que maneje casos de Internet.

Procedimientos de Recolección de Evidencias

Los métodos empleados en el proceso de recolección de pruebas se deberán detallar tanto como fuera posible. Éstos metodos se caracterizaran por no ser ambiguos y por minimizar el número de decisiones que es necesario tomar durante la recolección de evidencias. **Transparencia** Los métodos empleados para la recolección de evidencias deberán ser transparentes y reproducibles.

Pasos para la recolección de evidencias

- Listar los sistemas que se vieron involucrados en el incidente y especificar de cuáles se obtendrón las evidencias.
- Establecer qué es lo que se entiende como relevante y admisible.
- Obtener el orden pertinente de volatilidad para cada sistema.
- Eliminar las conexiones externas de los sistemas.
- Siguiendo el orden de volatilidad, recoger las evidencias usando las herramientas oportunas.
- Registrar la hora del reloj del sistema.
- Documentar cada paso dado.
- Tomar nota de quién estaba allá y qué estaba haciendo, qué es lo ha observado y cómo ha reaccionado.
- Hasta donde sea posible, firmar digitalmente las evidencias obtenidas, siempre que esto no altere el contenido de las propias evidencias.

3.5.4. Procedimiento de Almacenamiento

Las evidencias deben ser estrictamente guardadas. Y los responsables de seguridad deben quedar perfectamente identificados en un documento destinado para tal efecto.

Custodia

Se deberán documentar los siguientes hechos:

- Dónde, cuándo y por quién fueron descubiertas y recolectados las pruebas.
- Dónde, cuándo y por quién fueron manejadas o examinadas las pruebas.
- Quién ha custodiado las pruebas y durante qué periodo. Cómo han sido almacenadas.
- Cuando cambiarán de custodia las pruebas, cuándo y cómo se hizo la transferencia.

Dónde y cómo almacenar las pruebas

Si es posible, se deben utilizar medios de almacenamiento comunes y conocidos.

El acceso a las pruebas deberá ser extremadamente restringido, y estar claramente documentado. Deberá ser posible detectar accesos no autorizados.

Herramientas necesarias

Los programas que se utilizan para la recolección de pruebas y el análisis forense deberáan almacenarse en dispositivos de sálo lectura (como por ejemplo, un CD).

Entre las herramientas utilizadas se deberá incluir:

- Un programa para examinar procesos (por ejemplo, 'ps')
- Programas para examinar el estado del sistema (por ejemplo, 'ifconfig', 'netstat'...)
- Un programa para hacer copias a nivel de bit (por ejemplo, 'dd')

- Scripts que automaticen la recolección y el análisis de las evidencias (por ejemplo, TCT).
- Un programa que permita la migración de datos del sistema atacado a otro remoto (por ejemplo, 'nc') .

3.6. Caso de Estudio

El caso de estudio que se presenta aquí corresponde con el análisis de una intrusión real, en él se irán explicando los pasos que se deben dar para conseguir realizar un análisis completo y eficaz.

Desde un punto de vista práctico, una vez que se ha detectado (o se sospecha) que la integridad del sistema ha podido ser vulnerada por algún tipo de intrusión. Este estudio lo puedes ver detallado en el anexo de analisis forense.

Capítulo 4

Reforma de Seguridad

Haré un análisis de las necesidades de servicios de red y su implementación en sistema de cómputo, aislado dentro de un sitio, cuya actividad es restringida a un grupo de personas, donde, la única manera a acceder a algun tipo de información será a través de los servicios de red. Para esto, nos ayudaremos de herramientas de software y hardware, que pueden ser libres o de pago de licencia.

En el capito anterior, se realizo un estudio de análisis forense, en el cual nos da como resultado la confiabilidad del sistema o de los archivos, esto con base de las bitácoras, las herramientas, etc. No descarte la peor de las posibilidades, no encontrar nada o desconfiar de los respaldos.

Sin olvidar, dado el resultado dado por el análisis, se deberá de colocar especial atención en aquellos puntos de vulnerabilidad. Las consideraciones deberán de surgir dada la posible condición hostil en Internet, al igual que en cualquier sistema informacion, el objetivo es minimizar las posibilidades de siniestros y en caso que ocurran, tener un apartado en la política de recuperación.

4.1. Preinstalción

Ante la situación de instalar un servidor "confiable", se deberá de hacer un análisis:

- Quienes van a interactuar en el sistema.

- Que políticas de acceso.
- Que servicios se prestaran.
- Cada cuando se monitoreara el sistema.
- Mantenimiento del equipo.
- Uso de redes locales

Hay al menos, tres niveles de protección que se deben implementar :

Seguridad de Red: se protege toda la red de computadoras conectadas a Internet, colocando una barrera de acceso que actúa de pared o muro de entrada denominado FIREWALL.

Seguridad de computadora: se implementa seguridad sobre cada una de las computadoras conectadas a Internet y se definen conforme sea el sistema operativo con la que opere.

Seguridad de transacción: es aplicable a cada una de las aplicaciones que se desarrollan para trabajar en Internet. Consiste en el logeo de contraseñas de accesos (password), mecanismos de encriptación de la información, etcétera.

4.1.1. Políticas de Seguridad de Cómputo

Como se describio en el capitulo 1, es el documento publico para los administrador y usuarios donde se establecen las normativas de uso

, deben de conocer las políticas de seguridad. Implementadas, estas políticas deben de ser únicas entendibles y de fácil acceso. Donde deberán aceptar los involucrados directamente y aceptadas por los directivos, para obtener el respaldo en caso de que hayan sido quebrantadas. En caso de que no existiesen deben de hacerse por el motivo de que todo va regido a estas, tanto derechos como obligaciones de usuarios y administrador(es), reglas de firewall, etc. En caso de que existieran hacerlas cumplir, esto es, si se excluyen personal al área de servidores y no existe este control, hacerlo valer y actualizar las PSC, esto en ámbito físico, con respecto al servidor si mencionamos no conectarse por `telnet` y el demonio esta activo no estamos cumpliendo con el requerimiento y están fracturando la seguridad. Se deberá evaluar el tipo de sistema operativo que se va ha reinstalar y si las necesidades cubren el ya establecido.

Esto nos lo va ha regir el tipo de servicios y aplicaciones que se otorgan a las redes tanto local como publica. Las políticas como la economía de la empresa.

Crear una política simple y genérica para su sistema, para que sus usuarios la puedan entender con facilidad y seguir. Esto debería proteger los datos que pretende poner a salvo y también la privacidad de sus usuarios. Algunos detalles que tiene que considerar adicionalmente quienes tienen acceso al sistema, a quien se le permite instalar programas en el sistema, quien es el propietario de determinados datos, métodos de recuperación de pérdidas y uso apropiado del sistema.

Una política de seguridad generalmente aceptada empieza con esta frase:
"Todo lo que no está permitido, está prohibido"

Esto significa que, salvo que garantice acceso a un servicio para un usuario, el usuario no debería usar ese servicio hasta que no le ofrezca el acceso. Esté seguro de que las políticas funcionan en su cuenta regular de usuario, diciendo "Ah, no tengo permiso para resolver este problema, lo tendré que hacer como root o administrador", le puede llevar a descubrir agujeros de seguridad que son bastante obvios, e incluso otros que no han sido utilizados aun.

Se deberá evaluar la adquisición del sistema operativo, si existe una versión mas actualizada y que desventajas presenta o mejoras a esta previa. Tendremos que recabar información de distintos distribuidores del sistema operativo para nuestro servidor, teniendo presente el tipo de plataforma de nuestra estación de trabajo. Como la evolución de sitios que se dedican a la revisión de herramientas y sistemas operativos, como por ejemplo:

Actualizaciones de nuestro sistema Tendremos que instalar lo necesario y no programas que no se van a utilizar

Servicio que se prestan Solo utilizar los puertos que sean necesarios "solo los que se le entrega servicio"

Para mayor referencia revise los capitulos 1 y 2 como las direcciones de web antes mencionadas para PSC.

4.1.2. Niveles de Acceso

Los programas de control de acceso deberán identificar a los usuarios autorizados a usar determinados sistemas, con su correspondiente nivel de acceso. Las distinciones que existen en los niveles de acceso están referidos a la lectura o modificación en sus diferentes formas. De acuerdo a ello se tienen los siguientes niveles de acceso a la información:

- Nivel de consulta de la información no restringida o reservada. El privilegio de lectura esta disponible para cualquier usuario y sólo se requiere un conocimiento de la estructura de los datos, o del Sistema de otro usuario para lograr el acceso. La autorización de lectura permite leer pero no modificar la base de datos.

- Nivel de mantenimiento de la información no restringida o reservada. El concepto de mantenimiento de la información consiste en:

  Ingreso. Permite insertar datos nuevos pero no se modifica los ya existentes.

  Actualización. Permite modificar la información pero no la eliminación de datos.

  Borrado. Permite la eliminación de datos.

- Nivel de consulta de la información incluyendo la restringida o reservada. Un usuario puede tener asignados todos, ninguno o una combinación de los tipos de autorización anteriores. Además de las formas de autorización de acceso de datos antes mencionadas, es posible autorizar al usuario para que modifique el esquema de la base de datos, pero es preferible que esta función sea de responsabilidad del Centro de cómputo.

- Nivel de mantenimiento de la información incluyendo la restringida. Cada palabra clave debe tener asignado uno de los niveles de acceso a la información mencionada anteriormente. La forma fundamental de autoridad es la que se le da al administrador de la base de datos, que entre otras cosas puede autorizar nuevos usuarios, reestructurar la base de datos, etc. Esta forma de autorización es análoga a la que se provee a un "súper usuario" o al operador para un sistema operativo.

La administración de mantener al sistema seguro y hacerlo se divide en dos diferentes niveles:

- Nivel usuario

- Nivel sistema

Las consideraciones para estos dos niveles son muy parecidas, pero se logran de manera muy diferente. El lograr la seguridad a nivel usuario (que, a fin de cuentas, es lo que los usuarios requieren de nosotros) implica haber logrado plenamente la seguridad a nivel sistema. Los principales a incluir la seguridad en la administración son:

Confidencialidad A nivel usuario, la confidencialidad significa que nadie que no está explícitamente autorizado pueda tener acceso a los datos privados de cada usuario. A nivel sistema es un poco más ambiciosa: Nadie que no sea el administrador debe tener acceso a la información de configuración y funcionamiento del sistema. Si un atacante quiere penetrar nuestro sistema, lo primero que debe hacer es reunir cuanta información le sea posible antes de hacerlo -?Qué sistema operativo tenemos instalado? Qué programas utilizamos para brindar cada uno de los servicios? Qué servicios tenemos que sean más fáciles de explotar? Cómo tenemos configurado cada uno de estos servicios? Si logramos negarle el acceso a esta información, no sólo nos evitaremos dolores de cabeza de varios días de duración en lo que reinstalamos todo (medida indispensable cuando la seguridad de nuestro sistema ha sido comprometida), sino que garantizaremos la confidencialidad a nivel usuario.

Consistencia La consistencia significa que todo funcione como debería funcionar, que todo está donde le dejamos, que no haya nada en el sistema que nos llame especialmente la atención. Para un usuario, esto puede significar que algún archivo suyo, su correo o su pagina Web hayan sido alterados o simplemente borrados. A nivel sistema, esto es mas grave. En un sistema Unix típico hay miles de archivos, incluyendo binarios, bibliotecas, archivos de configuración, archivos de datos, archivos de los usuarios. Y la modificación de cualquiera de estos no puede significar mas que problemas. Muchas veces, el atacante lo hará con el objetivo de:

- Mantener una puerta secreta abierta para permitirle acceso en el futuro. Para lograrlo, las modificaciones son normalmente a binarios (programas) del sistema, aunque muchas veces son archivos de configuración.
- Capturar información confidencial de los usuarios. Para esto normalmente modifican un binario de uso frecuente (por ejemplo, `ls`, que despliega los contenidos de un directorio, o el popular lector de correo pine).
- Dañar la imagen de la corporación. La manera mas común de hacerlo es modificando la pagina principal o alguna sub-pagina para que de información ilegítima.

4.1.3. Control de acceso Fisico y Logico

El acceso al área de sistemas puede crear un significativo problema de seguridad, sino se va a delimitar en donde se van ha colocar las áreas de punto critico como nuestra área de

cómputo. El acceso normal debe ser dado solamente a la gente que regularmente trabaja en esta área. Cualquier otra persona, de otro modo puede tener acceso únicamente bajo control estricto.

Para mantener la seguridad física de su área de sistema es su primera línea de defensa. Para ello deberá tomar en consideración el valor de sus datos, el costo de protección, el impacto que su pérdida podría tener en su organización y la motivación, competencia y oportunidades de la gente que podría querer dañar los datos o el sistema.

Acceso limitado a los terminales o consolas del sistema

Los terminales que son dejados sin protección pueden ser mal usados. Cualquier terminal que puede ser utilizado como acceso a los datos de un Sistema controlado, debe ser encerrado en un área segura o guardado, de tal manera que no sean usados, excepto por aquellos que tengan autorización para ello. Igualmente, se deberá considerar la mejor manera de identificar a los operadores de terminales del Sistema, y el uso de contraseñas, cuando un terminal no sea usado pasado un tiempo predeterminado (5 - 10 minutos). Restricciones que pueden ser aplicadas:

- Determinación de los períodos de tiempo para los usuarios o las terminales.
- Designación del usuario por terminal o del terminal por usuario.
- Limitación del uso de programas para usuario o terminales.
- Límite de tentativas para la verificación del usuario.
- Tiempo de validez de las trasmisiones.

Control de acceso a la información Algunos usuarios o extraños (personal no autorizado) pueden encontrar alguna forma mediante la cual, logren el acceso al sistema o la base de datos y descubrir información clasificada o datos no autorizados. Se deberá considerar la existencia de:

Programas de Control. Deben existir programas protegidos que mantengan y controlen a los usuarios y sus derechos de acceso, ya sea por grupos o individualmente. El uso de tal programa puede conferir al usuario algunos de los privilegios que corresponden al controlador de dichos programas. La transferencia de privilegios es adecuada si el programa actúa como filtro de la información.

Palabra de Acceso (Password). Es una palabra o código que debe teclearse al sistema de computadora antes de generar un proceso. Constituye un procedimiento de seguridad que protege los programas y datos contra los usuarios no autorizados. La identificación de un individuo debe ser muy difícil de imitar y copiar. Aunque su nombre pueda ser único, es fácil que cualquiera que observe a quienes tienen acceso al sistema lo copie, por lo que no es una clave adecuada. Una vez que se obtiene una clave de acceso al sistema, ésta se utiliza para entrar al sistema de la base de datos desde el sistema operativo. La responsabilidad del manejo de la clave corresponde tanto al que accede al sistema operativo.

A fin de proteger el proceso de obtención de una llave del sistema, cuando el usuario realiza la entrada (en inglés LOGIN), solicita una clave de acceso con el nombre del usuario, la cual consiste de unas cuantas letras elegidas por el usuario.

Un intruso puede intentar descubrirla de dos maneras: una, observando el ingreso de la clave y otra, utilizando un método de ensayo y error para introducir posibles claves de acceso y lograr entrar. El sistema de computación debe cerrarse después que un individuo no autorizado falle dos veces, por ejemplo, al intentar ingresar una clave de acceso. Las claves de acceso no deben ser largas puesto que son más difíciles de recordar. En todo proceso corporativo e institucional es recomendable que el responsable de cada área asigne y actualice en forma periódica el password a los usuarios.

No se puede depender de que la ausencia de un operador o responsable de un servidor trabe la operatividad normal de una institución, por lo que puede ser necesario el establecimiento de un procedimiento de tener un duplicado de los passwords asignados, bajo un esquema de niveles jerárquicos, en un sobre lacrado. Esto es, el Jefe Inmediato superior tendrá en un sobre lacrado, los passwords de su personal, debiendo utilizar un cuaderno de control, cuando exista la necesidad de romper el sobre lacrado (anotando fecha, hora, motivo, etc.), así como un procedimiento de cambio de passwords periódicos y por dichas eventualidades.

4.2. Estrategias de recuperación

Se debe de definir los procedimientos y planes de acción para el caso, de que suceda o ocurra una posible falla, siniestro o desastre en el área de Informática, se deberán considerar, todas las áreas de los usuarios que procesan o almacenan la información por medio de los sistemas de computo.

Cuando ocurra una contingencia, se deberá conozca a detalle el motivo que la originó y el daño producido, lo que permitirá recuperar en el menor tiempo posible el proceso perdido

y no remediar el daño. Los procedimientos deberán ser de ejecución obligatorias y bajo la responsabilidad de los encargados de la realización de los mismos, es tener procesos de verificación de su cumplimiento de los procesos. En estos procedimientos estará involucrado todo el personal de la Institución. Los procedimientos de planes de recuperación de desastres deberán emanar de la máxima autoridad Institucional, para garantizar su difusión y su estricto cumplimiento. Las actividades a realizar en un Plan de Recuperación de Desastres se pueden clasificar en tres etapas:

4.2.1. Actividades Previas al Desastre

Son todas las actividades de planeación, preparación, entrenamiento y ejecución de las actividades de resguardo de la información, que nos aseguren un proceso de recuperación con el menor costo posible. Podemos detallar las siguientes actividades generales :

1. Establecimiento del Plan de Acción.
2. Formación de Equipos Operativos.
3. Formación de Equipos de Evaluación (auditoría de cumplimiento de los procedimientos en materia de seguridad).

Establecimiento de Plan de Acción

En esta fase de planeamiento se deberán establecer los procedimientos relativos a: Sistemas e Información. La Institución deberá tener un inventario de todo equipo de computo y sus características que juega en la organización. Los Sistemas de Información deberá detallar los siguientes datos:

Nombre del Sistema.

Lenguaje o Paquete con el que fue creado el Sistema. Programas que lo conforman (tanto programas fuentes como programas objetos, rutinas, macros, etc.).

La Dirección (Gerencia, Departamento, etc.) que genera la información base (el dueño del Sistema).

Las unidades o departamentos (internos/externos) que usan la información del Sistema.

El volumen de los archivos con que trabaja el Sistema.

El volumen de transacciones diarias, semanales y mensuales que maneja el sistema

El equipo necesario para un manejo sétimo del Sistema.

La(s) fecha(s) en las que la información que con carácter de urgencia.

Actividades a realizar para volver a contar con el Sistema de Información (actividades de Restaruración).

Con toda esta información se deberá realizar una lista de prioridades (un ranking) de los Sistemas de Información necesarios para que la Institución pueda recuperar su operatividad perdida en el desastre (contingencia).

Equipos de cómputo

Se debe realizar un inventario actualizado de los equipos de manejo de información (computadoras, impresoras, etc.), especificando su contenido (software utilizado y su origen, principales archivos que contiene), su ubicación y nivel de uso Institucional.

Como parte de la protección de los Activos Institucionales se deben contratar pólizas de seguros comerciales, deberá realizar en el contrato de seguro, que en casos de siniestros, la restitución del equipo dañado se podrá hacer por otro de mayor potencia (por actualización tecnológica), siempre y cuando está dentro de los montos asegurados.

Señalización o etiquetado de los Computadores de acuerdo a la importancia de su contenido, para ser priorizados en caso de evacuación. Por ejemplo etiquetar (colocar un sticker) de color rojo a los Servidores, color amarillo a las PC's con Información importante o estratégica y color verde a las PC's de contenidos normales.

Tener siempre actualizada una relación de PC's requeridas como mínimo para cada Sistema permanente de la Institución que por sus funciones constituyen el eje central de los Servicios Informáticos de la Institución, las funciones que realizaría y su posible uso en dos o tres turnos de trabajo, para cubrir las funciones básicas y prioritarias de cada uno de estos Sistemas.

Respaldos

Obtención y almacenamiento de los Respaldos de Información son los llamados **BACKUPS**, se deberán establecer los procedimientos para la obtención de copias de seguridad de todos

los elementos de software necesarios para asegurar la correcta ejecución de los Sistemas o aplicativos de la Institución. Para lo cual se debe contar con :

1. Backups del sistema operativo, se deberá de tener al menos varios copias de los sistemas operativos y las versiones que se manejan. especificando origen, vendedor, medio de traslado, compra realizada por que persona y su puesto.

2. Backups del software base paquetes y/o Lenguajes de Programación con los cuales han sido desarrollados o interactúan nuestros aplicaciones en las instituciones).

3. Backups de aplicaciones de terceros. Se deberá tener el código impreso y en disco de cualquier programa hecho para la empresa, se deberá expresar quien lo realizo en que periodo en la historia de la empresa.

4. Backups de los Datos (Bases de Datos, índices, tablas de validación, passwords, y todo archivo necesario para la correcta ejecución del software de aplicación de nuestra Institución).

5. Backups del Hardware. Se puede implementar bajo dos modalidades.:

   - Modalidad Externa. Mediante convenio con otra Institución que tenga equipos similares o mayores y que brinden la seguridad de poder procesar nuestra Información, y ser puestos a nuestra disposición, al ocurrir una contingencia mientras se busca una solución definitiva al siniestro producido. Este tipo de convenios deberá considerar el equipo como del entorno y facilidades de trabajo que cada institución se compromete a brindar, y deberá ser actualizado cada vez que se efectúen cambios importantes de sistemas que afecten a cualquiera de las instituciones.
   - Modalidad Interna. Si tenemos mas de un local, en ambos debemos tener señalados los equipos, que por sus características técnicas y capacidades, son susceptibles de ser usados como equipos de emergencia del otro local, debiéndose poner por escrito (igual que en el caso externo), todas las actividades a realizar y los compromisos asumidos.

   En ambos casos se deberá probar y asegurar que los procesos de restauración de Información hagan posible el funcionamiento adecuado de los sistemas. En algunos casos puede ser necesario volver a recompilar nuestro software aplicativo bajo plataformas diferentes a la original, por lo que es imprescindible contar con los programas fuentes, al mismo grado de actualización que los programas objeto.

6. Políticas (Normas y Procedimientos de Backups) Se deben establecer los procedimientos, normas y determinar responsabilidades en la obtención de los Backups mencionados anteriormente en el punto anterior, debiéndose incluir:

   - Periodicidad de cada tipo de backup. Respaldo de información de movimiento entre los períodos que no se sacan Backups (backups incrementales).
   - Uso obligatorio de un formulario estándar para el registro y control de los Backups. Dara de especificar fecha maquina respaldo previo.
   - Correspondencia entre la relación de Sistemas e Informaciones necesarias para la buena marcha de la empresa, y los backups efectuados.
   - Almacenamiento de los backups en condiciones ambientales sétimas, dependiendo del medio magnético empleado.
   - Reemplazo de los backups, en forma periódica, saber cuantas veces se puede hacer respaldos antes que el dispositivo se deteriore.
   - Almacenamiento de los Backups en locales diferentes donde reside la información primaria (evitando la pérdida si el desastre alcanzo todo el edificio o local estudiado).
   - Pruebas periódicas de los Backups (Restauración del sistema), verificando su funcionalidad, a través de los sistemas, comparando contra resultados anteriores confiables.

Formación de Equipos Operativos

En cada unidad operativa de la institución deberá designarse un respaldo, que sirva de enlace para el almacenamiento de información y sirva para la operatividad Institucional se deberá designar un responsable de la seguridad de la Información de su unidad. Pudiendo ser el jefe de dicha área Operativa. Sus labores serán:

- Ponerse en contacto con los propietarios de las aplicaciones y trabajar con ellos.
- Proporcionar soporte ténico para las copias de respaldo de las aplicaciones.
- Planificar y establecer los requerimientos de los sistemas operativos en materia de archivos, bibliotecas, utilitarios, etc., para los principales sistemas y subsistemas.
- Supervisar procedimientos de respaldo y restauración.

- Supervisar la carga de archivos de datos de las aplicaciones, y la creación de los respaldos incrementales.
- Coordinar líneas, terminales, módems y otros aditamentos para comunicaciones.
- Establecer procedimientos de seguridad en los sitios de recuperación.
- Organizar la prueba de hardware y software.
- Ejecutar trabajos de recuperación.
- Cargar y probar archivos del sistema operativo y otros sistemas almacenados en el local alterno.
- Realizar procedimientos de control de inventario y seguridad del almacenamiento en el local alterno.
- Establecer y llevar a cabo procedimientos para restaurar el lugar de recuperación.
- Participar en las pruebas y simulacros de desastres.

Formación de Equipos de Evaluación (auditoría de cumplimiento de los procedimientos sobre Seguridad)

Esta función debe ser realizada de preferencia por personal de auditoría o seguridad, de no ser posible la realizara el personal del área de Informática, debiendo establecerse claramente sus funciones, responsabilidades y objetivos :

- Revisar que las normas y procedimientos con respecto a backups y seguridad de equipos y datos se cumpla.
- Supervisar la realización periódica de los backups, por parte de los equipos operativos, comprobando físicamente su realización, adecuado registro y almacenamiento.
- Revisar la correlación entre la relación de Sistemas e Informaciones necesarios para la buena marcha de la Institución, y los backups realizados.
- Informar de los cumplimientos e incumplimientos de las Normas, para las acciones de corrección respectivas.

4.2.2. Actividades Durante el Desastre

Una vez presentada la contingencia o siniestro, se deberán ejecutar las siguientes actividades, planificadas previamente:

Plan de Emergencias

En este plan se establecen las acciones se deberán realizarse cuando se presente un siniestro, así como la difusión de las mismas. Es conveniente prever los posibles escenarios de como ocurriría el siniestro :

- Durante el día.
- Durante la Noche o madrugada.

Este plan deberá incluir la participación y actividades a realizar por todas y cada una de las personas que se encuentren presentes en el área donde ocurre el siniestro, debiendo detallar : Vías de salida o escape. ya sea físico o lógico.

Plan de Evacuación del Personal.

Plan de puesta a buen recaudo de los activos (incluyendo los activos de Información) de la Institución (si las circunstancias del siniestro lo hacen posible)

Ubicación y séquela de los elementos contra el siniestro (extinguidores, cobertores contra agua, etc.) Secuencia de llamadas en caso de siniestro, tener a la mano: elementos de iluminación (linternas), lista de teléfonos de Bomberos /Ambulancia, Jefatura de Seguridad y de su personal (equipos de seguridad) nombrados para estos casos.

Formación de Equipos

Establecer claramente cada equipo (nombres, puestos, ubicación, etc.) con funciones claramente definidas que deberá ejecutar durante el siniestro. Si bien la premisa básica es la protección de la Integridad del personal, en caso de que el siniestro lo permita, deberá de

existir dos equipos de personas que actúen directamente durante el siniestro, un equipo para combatir el siniestro y otro para el salvamento de los recursos informáticos, de acuerdo a los lineamientos o clasificación de prioridades, para salvar los equipos séquela en las políticas.

Entrenamiento

Establecer un programa de practicas periódicas de todo el personal en la lucha contra los diferentes tipos de siniestros, de acuerdo a los roles asignados en los planes de evacuación del personal o equipos, para minimizar costos se puede aprovechar fechas de recarga de extinguidores, charlas de los proveedores, etc. Un aspecto importante es que el personal tome conciencia de que los siniestros (incendios, inundaciones, terremotos, apagones, etc.) pueden realmente ocurrir, y tomen con seriedad y responsabilidad los entrenamientos, por eso es conveniente que participen los directivos, dando el ejemplo de la importancia que la dirección otorga a la Seguridad Institucional.

4.2.3. Actividades Después del Desastre

Después de ocurrido el Siniestro o Desastre es necesario realizar las actividades que se detallan a continuación, las cuales deberán estar especificadas en el Plan de Acción elaborado en el punto.

Inmediatamente después que el siniestro ha terminado, se deberá evaluar la magnitud del daño producido, qué sistemas esta(n) afectado(s), qué equipos han quedado sin operatición, cuales se pueden recuperar, y en cuanto tiempo, etc. Adicionalmente se deberá lanzar un pre-aviso a la Institución con la cual tenemos el convenio de respaldo, para avanzar en las labores de preparación de entrega de los equipos por dicha Institución.

Priorización de actividades del Plan de Acción

Toda vez que el plan de acción es general y contempla una pérdida total, la evaluación de daños reales y su comparación contra el plan, nos dará la lista de las actividades que debemos realizar, priorizando siempre a las actividades estratégicas y urgentes de nuestra Institución.

Es importante evaluar la dedicación o el trabajo del personal a actividades que puedan no haberse afectado, intentar su asignar temporalmente las actividades afectadas, apoyar al

personal de los sistemas afectados y al soporte técnico.

Ejecución de Actividades

La ejecución de actividades implica la creación de equipos de trabajo para realizar las actividades previamente planificadas en el plan de acción. Cada uno de estos equipos deberá contar con un coordinador que deberá reportar diariamente el avance de los trabajos de recuperación y, en caso de producirse algún problema, reportarlo de inmediato a la jefatura a cargo del Plan de Contingencias.

Los trabajos de recuperación tendrán dos etapas: la primera la restauración del servicio usando los recursos de la Institución o local de respaldo.

Y la segunda etapa es volver a contar con los recursos en las cantidades y lugares propios del Sistema de Información, debiendo ser esta última etapa lo suficientemente rápida y eficiente para no perjudicar el buen servicio de nuestro Sistema e imagen Institucional y no perjudicar la operatividad de la Institución o local de respaldo.

Evaluación de Resultados

Una vez concluidas las labores de Recuperación del (los) Sistema(s) que fueron afectados por el siniestro, debemos de evaluar objetivamente todas las actividades realizadas, cómo se hicieron, qué tiempo tomaron, qué circunstancias se modificaron (aceleraron o entorpecieron) las actividades del plan de acción, cómo se comportaron los equipos de trabajo, etc.

De la evaluación de resultados y del siniestro, deberá salir dos tipos de recomendaciones; una la retroalimentación del plan de contingencias y otra una lista de recomendaciones para minimizar los riesgos y pérdidas ocasionados por el siniestro.

Retroalimentación del Plan de Acción

Con la evaluación de resultados, debemos de optimizar el plan de acción original, mejorando las actividades que tuvieron algún tipo de dificultad y reforzando los elementos que funcionaron adecuadamente. El otro elemento es evaluar cual hubiera sido el costo de no haber tenido nuestra Institución el plan de contingencias y llevarlo a cabo.

4.3. Análisis del sitio

Se deberá plantear desde su estructura del sitio conocer los planos y ubicación de instalaciones eléctricas y de red esto para contemplar cualquier intento de siniestro ya sea natural o provocado. Por ejemplo en donde estarán nuestros concentradores que no sean de fácil acceso y tener control absoluto de quien hace manipulaciones de cableado. Otro ejemplo será en donde estarán nuestras estaciones de trabajo o servidores como colocar los servidores hubs y en donde estará ubicado cada componente y así se deberá minimizar lo errores esto con el fin de mantener un entorno seguro. Se deberá analizar los permisos configuración e instalación de nuevo software para no arriesgar nuestras zonas de conflicto

Aseguramiento del sistema operativo

Se necesita saber donde está el equilibrio entre la facilidad de uso de su sistema y su seguridad. Por ejemplo, puede necesitar que cualquiera que llame por el módem a su sistema, nuestro módem tenga que devolver la llamada a su número de casa. Esto es más seguro, pero si alguien no está en casa, le hace más difícil conectarse. Podría poner su sistema Linux sin conexiones de red o a internet, pero esto le dificulta moverse por la web.

Si el sitio tiene posibilidades de mediano a gran alcance economico, se deberá establecer una "Política de Seguridad"que fije el nivel de seguridad que requiere ese sitio y que sistema de comprobación se realiza. Puede encontrar un ejemplo muy conocido de política de seguridad en http://ds.internic.net/rfc/rfc2196.txt. ha sido actualizado recientemente y contiene una gran estructura en la que basar la política de seguridad de su compañía.

La vulnerabilidad describe como de bien protegido está su equipo frente a otra red, y el potencial para alguien que pueda obtener acceso no autorizado. ¿Qúe está en juego si alguien entra en su sistema?.

¿Cuanto tiempo me llevaría recuperar/recrear cualquier dato que se ha perdido? Una inversión en tiempo ahora puede ahorrar diez veces más de tiempo con posterioridad si tiene que recrear los datos que se perdieron. ¿Ha verificado su estrategia de copias de respaldo, y ha verificado sus datos últimamente?

4.4. Medios de la Seguridad de su Sitio

Los intrusos tienen mucho tiempo a su disposición, y pueden evitar preguntarse como ha protegido su sistema con sólo intentar todas las posibilidades. Hay también varias razones más por las que un intruso puede estar interesado en su sistema, que se discutirá más tarde.

Seguridad del Host

Quizás el área donde se concentra más seguridad es en la seguridad basada en hosts. Típicamente implica estar seguro de que su sistema es seguro, y esperar que todas las demás personas de la red hagan lo mismo. Escoger buenas claves, asegurar los servicios de red de su host, mantener un buen registro de cuentas y actualizar los programas que tienen exploits conocidos que afectan a su seguridad, son las cosas de las que es responsable el administrador local. Aunque esto es absolutamente necesario, se puede convertir en una complicada labor si su red se hace muy grande.

Seguridad de su Red

La seguridad de la red es tan necesaria como la seguridad de un host. Con su simple sistema, o una red distribuida, internet o cientos, sino miles de equipos en la misma red, no puede tener la confianza de que todos esos equipos son seguros. Estando seguro de que sólo se le permite acceso a su red a usuarios autorizados, construyendo cortafuegos, usando cifrados fuertes, asegurándose de que no hay pícaros, o máquinas inseguras en su red, todo esto son parte de las preocupaciones y deberes del administrador de una red segura.

Seguridad mediante Oscuridad

Un tipo de seguridad que se debe discutir es "seguridad mediante oscuridad". Esto significa que haciendo algo como cambiar el nombre de conexión de 'root' a 'toor', por ejemplo, para intentar confundir a alguien que intenta entrar en sus sistema como root, es sólo una falsa sensación de seguridad, y puede originar unas consecuencias muy desagradables. Descanse seguro de que cualquier atacante de sistemas rápidamente se dará cuenta de esas mediadas de seguridad vacías. Por el hecho de que tenga un sitio pequeño o de relativamente bajo nivel de personalización, no significa que un intruso no esté interesado en lo que tiene. Discutiremos

en próximas secciones lo que está protegiendo.

4.5. Visión General

Lo primero que tenemos que tener en mente es que no existe nada como un sistema completamente seguro. Todo lo que puede hacer es aumentar la dificultad para que alguien pueda comprometer su sistema. En el caso medio del usuario de linux en casa, no se requiere demasiado para mantener alejado al cracker. Para usuarios con grandes requisitos (bancos, compañías de telecomunicaciones, etc) se requiere mucho más trabajo. Debido al valor de la información o a comprometer algun otro sistema de computo.

Otro factor a tener en cuenta es que cuanto más incremente la seguridad de su sistema, más intrusiva se vuelve la seguridad. Necesita decidir en que medida su sistema es utilizable y en que medida es seguro para sus propósitos. Por ejemplo, puede necesitar que cualquiera marque a su modem para que este devuelva la llamada a su casa. Esto es más seguro, pero si alguien no está en casa hace más difícil que se pueda conectar. También puede configurar su sistema linux sin conexión a internet, pero esto dificulta que pueda navegar por las webs. Si tiene un sitio medio-grande, debería establecer una "Política de Seguridad"que indique qué niveles requiere su sitio y qué medidas de evaluación se realizan.

4.5.1. Seguridad Local

Lo siguiente que tiene que observar es la seguridad de su sistema frente a los ataques de los usuarios locales. Una de las primeras cosas que un intruso intenta es obtener acceso como usuario local. Con una seguridad local laxa, pueden "actualizar"su acceso normal como usuario a acceso como root utilizando una serie de deficiencias (bugs) y configuraciones pobres de los servicios locales. Si está seguro de su seguridad local, el intruso tendrá otro obstáculo que saltar.

Los usuarios locales también pueden causar una serie de problemas de estragos con su sistema si son realmente quienes dicen que son. Proporcionar cuentas a personas desconocidas o con las que no tiene información de contacto es muy mala idea.

Antes de cambiar los permisos de cualquier sistema de archivos, esté seguro de que entiende lo que está haciendo. Nunca cambie los permisos de un archivo sólo porque parezca una forma fácil de hacer que las cosas funcionen. Determine siempre por qué el archivo tiene ese

permiso antes de cambiarlo.

Creación de nuevas cuentas

Debe estar seguro y proporcionar cuentas de usuario sólo con los requisitos mínimos para el trabajo. Si le crea una cuenta a su hijo, de 10 años, le puede interesar que sólo tenga acceso a un procesador de textos o un programa de dibujo, pero que no pueda hacer un `rm`(borrado del sistema de archivos en su modo total).

Aquí tiene varias buenas reglas para cuando permita a otros un acceso legítimo a su máquina:

Déle solo la mínima cantidad de privilegios que necesita.

Tenga cuidado con donde/cuando se conecta o si se debería conectar.

Esté seguro de eliminar aquellas cuentas que ya no son necesarias.

Muchas de las cuentas de usuarios locales que se usan para comprometer la seguridad de su sistema, son las que no han sido utilizadas en meses o años. Como nadie las usa proporcionan el vehículo ideal para un ataque.

Seguridad del root

Otros atacantes locales frecuentemente es el administrador del sistema linux. (Sí, Usted el administrador.). Recuerde que sólo debe usar la cuenta de root para tareas específicas y cortas y el resto hacerlo como usuario normal. Ejecutar como root todo el tiempo es una acto de alto riesgo. Use su o sudo para realizar las tareas que necesite.

Varios trucos para evitar que pueda estropear su propio Unix como root:

Cuando vaya a ejecutar un comando complejo, intente primero ejecutarlo de una forma no destructiva, especialmente los comandos con comodines. Por ejemplo, si quiere hacer un `rm foo*.bak`. haga primero : `ls foo*.bak` y compruebe realmente los archivos que va a borrar. También puede usar `echo` en algunas circunstancias. Algunas personas encuentran útil hacer un `touch /-i` en sus sistemas. Esto hará que comandos como `rm -rf *` le pregunten primero si realmente quiere borrar todos esos archivos. (Esto

lo hace la shell, resolviendo primero el `-i` y tratándolo como la opción `-i` de `rm`). Esto no nos será útil con comandos `rm` sin *.

El comando path es importante para el usuario root. El comando path o la variable de entorno PATH define la busqueda de programas por la shell. Intente limitar tanto como sea posible el comando path para el usuario root, y nunca use ".", (directorio actual) en su PATH. Además, no tenga directorios con permiso de escritura en su ruta de búsquedas, ya que esto puede permitir a los atacantes modificar o poner nuevos binarios que se pueden ejecutar como root la próxima vez que ejecute el comando.

Nunca use la suit de herramientas rlogin/rsh/rexec (llamadas las r-utilidades) como as root. Puede ser objeto de diversos tipos de ataques y es peligroso ejecutarlas como root. Nunca cree un archivo .rhosts para root. El archivo `/etc/securetty` contiene la lista de terminales desde las cuales se puede conectar el root. Por defecto (en Linux RedHat) se incluyen sólo las consolas virtuales (vtys). Tenga mucho cuidado al añadir otra cosa a este archivo. Debería poder presentarse al sistema de forma remota como un usuario normal u entonces usar su si lo necesita (mejor sobre ssh u otro canal cifrado) con lo cual no tiene necesidad de entrar directamente como root. Cambie a root sólo para realizar tareas específicas simples. Si tiene la necesidad de hacer algo, inténtelo en una shell de usuario normal hasta que esté seguro de qué hay que hacer como root. Actúe de forma lenta y meditada cuando sea root. Sus acciones podrían afectar a un montón de cosas. ¡Piénselo antes de antes de teclear!.

Si de forma inevitable tiene que permitir a alguien hacerse root en su máquina, hay algunas herramientas que le pueden ayudar. Sudo permite a los usuarios utilizar sus claves para acceder a un limitado número de comandos como root. Esto le permitiría, por ejemplo, permitir a un usuario montar y desmontar dispositivos removibles en su linux, pero sin tener otros privilegios. sudo también mantiene un registro de todos los intentos y éxitos, permite seguir la pista de los usuarios con los comandos y quor esta razón sudo funciona bien [illegible] luso en situaciones donde un determinado número de usarios tienen acceso de root, [illegible]ro usan sudo para que se pueda tener un rastro de los cambios realizados.

4.5.2. Archivos y Seguridad del Sistema de Archivos

Al tener algunos minutos de preparación y planificación antes de conectar su sistema a producción o servicio le puede ayudar a proteger su sistema y los datos que tiene almacenados en él.

No debería haber ninguna razón para que los directorios home de los usuarios permitieran que programas SUID/SGID se ejecutaran desde allí. Use la opción `nosuid` en `/etc/fstab` para las particiones que tienen permiso de escritura por alguien distinto de root. También le puede interesar usar `nodev` y `noexec` en las particiones de los directorios home de los usuarios. También en `/var`, lo que prohibe la ejecución de programas y creación dispositivos de bloque o carácter, que en ningún caso serían necesarios.

Si exporta sistemas de archivos vía NFS, esté seguro de configurar `/etc/exports` con los accesos lo más restrictivos posibles. Esto significa no usar plantillas, no permitir acceso de escritura a root y montar como sólo lectura siempre que sea posible.

Configura la máscara de creación de archivos para que sea lo más restrictiva posible. Son habituales 022, 033, y la más restictiva 077, y añadirla a `/etc/profile`. Ponga el límites al sistema de archivos en lugar de `ilimitado` como está por defecto. Puede controlar el límite por usuario utilizando el módulo PAM de límite de recursos y `/etc/pam.d/limits.conf`. Por ejemplo, los límites para un grupo 'users' podría parecer a esto:

```
@users     hard  core    0
@users     hard  nproc   50
@users     hard  rss     5000
```

Esto dice que se prohiba la creación de archivos core, restringe el número de procesos a 50, y restringe el uso de memoria por usuario a 5M.

Los archivos `/var/log/wtmp` y `/var/run/utmp` contienen los registros de conexión de todos los usuarios de su sistema. Se debe mantener su integridad ya que determinan cuando y de donde entró en su sistema un usario o un potencial intruso. Los archivos deberían tener los permisos 644, sin afectar a la normal operación del sistema.

El bit inmutable se puede usar para prevenir borrados accidentales o proteger un archivo para sobreescritura. También previene que alguien cree enlaces simbólicos a un archivo, que ha sido el origen de ataques basados en el borrados de los archivos /etc/passwd o /etc/shadow.

Los archivos `SUID` y `SGID` de su sistema son potenciales riesgos de seguridad y deberían ser controlados. Como estos programas garantizan privilegios especiales al usuario que los ejecuta, es necesario estar seguro que no hay instalados programas inseguros. Un truco favorito de los crackers es explotar programas con el bit SUID, y entonces dejar un programa SUID como puerta trasera para entrar la próxima vez, incluso aunque el agujero original ya esté tapado. Encuentre todos los programas SUID/SGID de su sistema y mantener la pista de lo que son,

para que esté prevenido de cualquier cambio que podría indicar un potencial intruso. Use el siguiente comando para localizar todos los progrmas SUID/SGID en su sistema:

`root# find / -type f` $-perm -04000 - o - perm - 02000$ Puede eliminar los permisos SUID o SGIG de un programa con `chmod`, y siempre puede devolverlo a su estado original si piensa que es absolutamente necesario.

Los archivos con permiso de escritura global, particularmente los del sistema, pueden ser un agujero de seguridad si un cracker obtiene acceso a su sistema y los modifica. Además los directorios con permiso de escritura global son peligrosos ya que permiten a un cracker añadir y borrar los archivos que quiera. Para localizar los archivos con permiso de escritura global, use el siguiente comando:

```
root# find / -perm -2 -print
```

Y esté seguro de saber por qué tienen esos permisos de escritura. En el curso normal de una operación archivos tendrán permisos de escritura, incluidos algunos de `/dev` y enlaces simbólicos.

Los archivos sin propietario también pueden ser un indicio de que un intruso ha accedido a su sistema. Puede localizar los archivos de su sistema que no tienen propietario o que no pertenecen a un grupo con el comando:

```
root# find / -nouser -o -nogroup -print
```

La localización de archivos `.rhosts` debería ser una de los deberes de la administración de su sistema regular, ya que estos archivos no se deberían permitir en sus sistema. Recuerde que un cracker sólo necesita una cuenta insegura para potencialmente obtener acceso a toda su red. Puede localizar todos los archivos .rhosts de su sistema con el siguiente comando:

```
root# find /home -name .rhosts -print
```

Finalmente, antes de cambiar permisos en cualquier sistema de archivos, esté seguro de que entiende lo que hace. Nunca cambie permisos de un archivo simplemente porque parezca la forma fácil de hacer una cosa. Siempre debe determinar porqué el archivo tiene esos permisos antes de modificarlos.

Estado de umask

El comando umask se puede usar para determinar el modo de creación de archivos por defecto en su sistema. Es el complemento octal a los modos de archivo deseado. Si los archivos se crean sin ningún miramiento de estado de permisos, el usuario, de forma inadvertida podrá asignar permisos de lectura o escritura a alguien que no debería tenerlo. De forma típica, el estado de umask incluye 022, 027 y 077, qque es lo más restrictivo. Normalmente umask se pone en `/etc/profile` y por tanto se aplica a todos los usuarios del sistema. Por ejemplo, puede tener una línea parecida a la siguiente:

```
# Pone el valor por defecto de umask del usuario
umask 033
```

Esté seguro de que el valor umask de root es 077, lo cual desactiva los permisos de lectura, escritura y ejecución para otros usuarios, salvo que explícitamente use chmod.

Permisos de archivos

Es importante asegurarse que sus archivos de sistema no los abren los usuarios o grupos que no tienen que realizar tareas de mantenimiento del sistema por ediciones casuales.

UNIX separa el control de acceso a archivos y directorios de acuerdo con tres características: propietario, grupo y otros. Siempre hay un sólo propietario, todos los miembros del grupo y cualquier otro.

Un resumen de los permisos Unix:

Propiedad: Qué usuario(s) y grupo(s) retiene el control de los permisos del nodo y del padre del nodo.

Permisos - Bits que se pueden fijar para permitir ciertos tipos de acceso a él. Los permisos para directorio pueden tener un significado diferente a los permisos para archivos.

Lectura (r):

Poder ver los contenidos de un archivo

Poder leer un directorio

Escritura (w):

Poder modificar o añadirle a un archivo

Poder borrar o mover archivos en un directorio

Ejecución(x):

Poder ejecutar un programa binario

Poder buscar en un directorio

Usted - El propietario del archivo

Group - El grupo al que pertenece

Everyone - Cualquiera del sistema

Save Text Attribute: (Para directorios) El sticky bit también tiene un significado diferente cuando se aplica a directorios. Si es sticky bit está activo en un directorio, entonces un usuario sólo puede borrar archivos que son propiedad del usuario o para los que tiene permiso explícito de escritura, incluso cuando tiene acceso de escritura al directorio. Esto está pensado para directorios como `/tmp`, que tienen permiso de escritura global, pero no es deseable permitir a cualquier usuario borrar los archivos que quiera. El sticky bit aparece como `t` en los listados largos de directorios.

Attributo SUID: (Para archivos) Este describe permisos al identificador de usuario del archivo. Cuando el modo de acceso de ID de usuario está activo en los permisos del propietario, y ese archivo es ejecutable, los procesos que lo ejecutan obtienen acceso a los recursos del sistema basados en el usuario que crea el proceso (no el usuario que lo lanza). Esto es causa de la utilización de muchos `buffer overflow`.

Atributo SGID: (Para archivos) Si está activo en los permisos de grupo, este bit controla el estado de "poner id de grupo"de un archivo. Actúa de la misma forma que SUID, salvo que afecta al grupo. El archivo tiene que ser ejecutable para que esto tenga algún efecto.

Atributo SGID: (Para directorios) Si activa el bit SGID en un directorio (con çhmod g+s directorio"), los archivos creados en ese directorio tendrán puesto su grupo como el grupo del directorio.

Usted - El propietario del archivo

Grupo - El grupo al que vd. pertenece

Otros - Cualquiera del sistema que no sea propietario o miembro del grupo.

```
Ejemplos:
  -rw-r--r--  1 cobian  users          114 Aug 28  1997 .zlogin
  1st bit - >directorio?              (no)
   2nd bit - >lectura por el propietario?          (si, por cobian)
    3rd bit -  por el propietario?         (si, por cobian)
     4th bit - >ejecución por el propietario?       (no)
      5th bit - >lectura por el grupo?          (sí, por users)
       6th bit - >escritura por el grupo?         (no)
        7th bit - >ejecución por el grupo?       (no)
         8th bit - >lectura por cualquiera?       (si, por cualquiera)
          9th bit - >escritura por cualquiera?      (no)
           10th bit - >ejecución por cualquiera?  (no)
```

Las siguientes líneas son ejemplos del conjunto mínimo de permisos que se requieren para llevar a cabo el acceso descrito. Puede querer dar más permisos que los listados, pero esto debería describir que hacen esos permisos mínimos sobre los archivos:

```
-r--------  Permite acceso de lectura al propietario
--w-------- Permite al propietario modificar o borrar el archivo
---x------  El propietario puede ejecutar este programa, pero no scripts de
                    shell que requieren permisos de lectura
---s------  Se ejecutará con usuario efectivo  ID = propietario
-------s--  Se ejecutará con usuario efectivo  ID = grupo
-rw------T No actualiza "instante de última modificación".  Normalmente
                    usado para archivos de intercambio (swap)
---t------  No tiene efecto.  (antes sticky bit)
```

Ejemplo de Directorio:

```
drwxr-xr-x  3 cobian  users          512 Sep 19 13:47 .public_html/
1st bit - >directorio?             (si, contiene muchos archivos)
 2nd bit - >lectura por el propietario?         (si, por cobian)
  3rd bit -  por el propietario?         (si, por cobian)
   4th bit - >ejecución por el propietario?       (sí, por cobian)
    5th bit - >lectura por el grupo?          (sí, por users)
     6th bit - >escritura por el grupo?         (no)
      7th bit - >ejecución por el grupo?        (sí, por users)
       8th bit - >lectura por cualquiera?        (si, por cualquiera)
        9th bit - >escritura por cualquiera?       (no)
         10th bit - >ejecución por cualquiera?  (sí, por cualquiera)
```

Las siguientes líneas son ejemplos del mínimo conjunto de permisos que se requieren para llevar a cabo el acceso descrito. Le puede interesar dar más permisos que los indicados, pero esto debería describir que hacen los permisos mínimos en los directorios:

```
dr--------  El contenido se puede listar pero no se pueden leer los
                        atributos.
d--x------  Se puede entrar en el directiorio y usado en las rutas de
                        ejecución completas.
dr-x------   Se pueden leer los atributos del archivo por el propietario.
d-wx------  Se pueden crear/borra archivos, incluso si no es el actual.
d------x-t  Previene el borrado de archivos por otros con acceso de
                       escritura. Usado en /tmp
d---s--s--  No tiene efecto
```

Los archivos de configuración del sistema (normalmente en /etc) es habitual que tengan el modo 640 (-rw-r—), y que sean propiedad del root. Dependiendo de los requisitos de seguridad del sistema, esto se puede modificar. Nunca deje algún archivo del sistema con permiso de escritura por un grupo o por cualquiera. Algunos archivos de configuración, incluyendo, /etc/shadow, sólo deberían tener permiso de lectura por root, y los directorios de /etc no deberías ser accesibles, al menos por otros.

SUID Shell Scripts. Los scripts de shell SUID son un serio riesgo de seguridad, y por esta razón el núcleo no los acepta. Sin importar lo seguro que piense que es su script de shell, puede ser utilizado para que un cracker pueda obtener acceso a una shell de root.

Verificar la integridad con Tripwire

Una forma cómoda de detectar ataques locales (y también de red) en sus sistema es ejecutar un programa que verifique la integridad como Tripwire. Tripwire ejecuta varios cheksums de todos los binarios importantes y archivos de configuración y los compara con una base de datos con valores de referencia aceptados como buenos. Así se detecta cualquier cambio en los archivos.

Es buena idea instalar tripwire en un medio de almacenamiento extraible para protegerlo físicamente. De esta forma no se puede alterar tripwire o modificar su base de datos. Una vez que tripwire se ha configurado, es buena idea ejecutarlo como parte de de los deberes habituales de administración para ver si algo ha cambiado.

Incluso puede añadir una entrada a crontab para ejecutar tripwire desde su disquete todas las noches y enviar por correo los resultados y verlos por la mañana, Algo como esto:

```
# set mailto
MAILTO=cobian
# run tripwire
15 05 * * * root /usr/local/adm/tcheck/tripwire
```

Le enviará por correo un informe cada mañana a las 5:15am. Tripwire detecta intrusos antes de que tenga otro tipo de noticias de ellos. Como son muchos los archivos que se modifican en su sistema, debería tener cuidado con lo que es la actividad de un cracker y lo que es la activiadad normal del sistema.

Caballos de Troya

El nombre de Caballo de Troya se debe al símil que se establece con los hechos contados por Homero en la Iliada. La idea es poner un programa binario, y consigue que otras personas lo carguen y lo ejecuten como root. Entonces puede comprometer la seguridad de su sistema mientras no se presta atención. Mientras se piensa que el binario que hemos lanzado hace algo (que puede estar bien), también puede estar saltándose la seguridad.

Debería tener cuidado con qué programas instala en su máquina. Las verificaciones son diversas por ejemplo md5 checksums y archivos firmados con pgp para pueda verificar que está instalando algo real. Jamás debería ejecutar un binario del cual no tiene las fuentes o sabe perfectamente qué programa es. Algunos atacantes también están publicando código fuente para que se pueda realizar un uso público y aprovecharse de ello.

Aunque esto puede ser complicado, debería asegurarse que obtiene las fuentes de un programa de el sitio real de su distribución. Si el programa se va a ejecutar como root, tendría que estar seguro que alguien de su confianza ha revisado y verificado el programa.

Seguridad de Claves y Cifrado (Encryption)

Unas de las más importantes características de seguridad usadas hoy son ls claves. Es importante para todos. usted y sus usuarios, tener claves seguras insospechadas. La mayoría de las distribuciones Unix recientes incluyen programas 'passwd' que no permiten a los usuarios poner claves fácilmente adivinables. Esté seguro de que su programa passwd está actualizado y tiene estas características.

Está fuera de los objetivos de este documento una discusión en profundidad sobre cifrado, pero daremos una introducción. El cifrado (encriptado) es muy útil, posiblemente incluso necesario en estos días. Hay distintos métodos de cifrar datos, cada uno con sus propios inconvenientes. Debería estar prevenido con algunos de los más comunes.

Cifrado de claves unix: La mayoría de los unix (y linux no es una excepción) usan DES (Data Encryption Standard) para cifrar sus claves. Estas claves cifradas se almacenan (típicamente) en `/etc/passwd` o (con menos frecuencia) en `/etc/shadow`. Cuando intenta presentarse al sistema, todo lo que teclea se cifra de nuevo y comparado con la entrada correspondiente del archivo `/etc/passwd`. Si coinciden, debe ser la misma clave y se le permite el acceso. DES en un algoritmo de cifrado de una sola vía. DES tiene reputación de débil en estos días de computadoraes rápidos. Ataques a la fuerza bruta, como crack o John de ripper (ver abajo), con frecuencia pueden adivinar claves salvo que su clave sea lo suficientemente aleatoria. Los módulos PAM (ver abajo) permiten usar rutinas de cifrado diferentes con sus claves (MD5 o parecidas).

PGP y Clave de Cifrado Pública (Public Key Cryptography)

Public Key Cryptography, tal y como se usa por PGP, implica un cifrado que usa una clave para cifrar y otra para descifrar. Tradicionalmente, el cifrado implica el uso de la misma clave para descifrar que la que se usó para cifrar. Esta clave privada"se debe conocer por ambas partes y de alguna forma, transferirse de uno a otro de forma segura.

La clave de cifrado pública alivia la necesidad de asegurar la transmisión de la clave usada para el cifrado usando dos claves separadas, una pública y otra privada. Cada clave pública de una persona está disponible por cualquiera para realizar el cifrado, mientras que a la misma vez cada persona mantiene su clave privada para descifrar el mensaje cifrado con la correspondiente clave pública.

Hay ventajas con las claves de cifrado públicas y privadas y puede informarse de las diferencias en las FAQ de RSA Cryptography indicadas al final de esta sección.

PGP (Pretty Good Privacy) está muy bien soportada por Linux. Las versions 2.6.2 y 5.0 se sabe que funcionan bien. Para iniciarse en PGP y como usarlo mire las PGP FAQ. http://www.pgp.com/service/export/faq/55faq.cgi Esté seguro de usar la versión que es aplicable a su país, ya que debido a las restricciones del gobierno de EE.UU, los cifrados fuertes se consideran armas militares y se prohibe su transferencia electrónica fuera del país.

También hay una guía paso a paso para configurar PGP en Linux, disponible en http://mercury.chem.pitt.ec Fue escrito para la versión Internacional de PGP. (pero es fácilmente adaptable a la versión EE.UU). Puede que necesite un parche para alguna de las últimas versiones de Linux, disponible en ftp://sunsite.unc.edu/pub/Linux/apps/crypto.

Podemos encontrar más indormación sobre cifrado en RSA cryptography FAQ, disponibles en http://www.rsa.com/rsalabs/newfaq/. Aquí encontrará información sobre términos como "Diffie-Hellman", "public-key cryptography", "Digital Certificates", etc.

SSL, S-HTTP, HTTPS y S/MIME

Con frecuencia los usuario preguntan las diferencias entre los distintos protocolos de seguridad y cifrado, y como se usan. Como esto no es un documento sobre cifrdo, es buena idea explicar brevemente qué es cada cosa y donde encontrar más información.

SSL: - SSL, o Secure Sockets Layer, es un método de cifrado desarrollado por Netscape para proporcionar seguridad en Internet. Soporta varios protocolos diferentes de cifrado y proporciona un servidor de verificación de clientes. SSL funciona a nivel de transporte, creando un canal seguro de datos cifrados y así puede cifrar datos de diversos tipos. Esto se ve con frecuencia cuando vamos a un sitio seguro para ver documentación en línea con Comunicatos y sirve como base para comunicaciones seguras con Comunicator, también como muchos Se puede encontrar más información en http://www.consensus.com/security/ssl-talk-faq.html. Otras implementaciones de seguridad de Netscape y un buen punto de partida para estos protocolos se puede encontrar en http://home.netscape.com/info/security-doc.html.

S-HTTP: - S-HTTP es otro protocolo que proporciona servicios de seguridad sobre internet. Fue diseñado para proporcionar servicios seguros a través de internet. Fue diseñado para proporcionar confidencialidad, autenticidad, integridad y no repudiabilidad [no se puede confundir con nadie] mientras que soporta mecanismos de gestión de múltiples claves y algoritmos de cifrado mediante opción de negociación entre las partes implicadas en cada transacción. S-HTTP está limitado al software específico que implementa y cifra cada mensaje individualmente [de RSA Cryptography FAQ, página 138]

S/MIME: - S/MIME, o Secure Multipurpose Internet Mail Extension. es un cifrado estándar utilizado para cifrar correo electrónico u otros tipos de mensajes sobre internet. Es un desarrollo estándar abierto de RSA, por tanto existe la esperanza de que con probabil-

idad lo veremos en Linux pronto. Se puede encontrar más información sobre S/MIME en http://home.netscape.com/assist/security/smime/overview.html.

Implementación Unix x-kernel IPSEC

Con CIPE. y otras formas de cifrado de datos, hay también una implementación de IPSEC para Unix o Linux. IPSEC es un trabajo de IETF para crear comunicaciones cifradas seguras a nivel de red IP, lo que proporciona autenticidad, integridad, control de acceso y cofidencialidad. Se puede encontrar información sobre IPSEC en http://www.ietf.org/html.charters/ipsec-charter.html. También puede encontrar enlaces con otros protocolos que implican gestión de claves y la lista de correo de IPSEC y sus archivos.

La implementación de Linux, que estásiendo desarrollada en la Universidad de Arizona, usa una estructura basada en objetos para implementar protocolos de red llamados x-kernel, y se puede encontrar en http://www.cs.arizona.edu/xkernel/hpcc-blue/linux.html. De forma más simple, x-kernel es u método de pasar mensajes al nivel del núcleo, que hace una más fácil implementación.

Como con las otras formas de cifrado, no se distribuye con el núcleo por defecto, debido a restricciones de exportación.

SSH (Secure Shell), stelnet

SSH y stelnet son programas que le permiten efectuar conexiones con sistemas remotos y tener una conexión cifrada.

SSH es un conjunto de programas usados como sustitución segura de rlogin, rsh y rcp. Usa claves públicas de cifrado para cifrar las comunicaciones entre dos hosts, también como para la verificación de claves. Esto se puede usar para asegurar las conexiones a un host remoto o copiar datos entre hosts, mientras que previene los ataques en el intermedio (secuestro de sesión) y falsificación de DNS (DNS spoofing). Realizará una conexión de los datos de su conexión y asegura las comunicaciones X11 entre hosts. La página principal de SSH, se puede encontrar en http://www.cs.hut.fi/ssh/

También puede usar SSH desde su estación de trabajo Windows hacia su servidor SSH Linux. Hay varias implementaciones de clientes Windows de libre disposición, incluyendo el que hay en http://guardian.htu.tuwien.ac.at/therapy/ssh/ y también implementaciones comerciales

de DataFellows, en http://www.datafellows.com/.

SSLeay es un implementación libre del protocolo Secure Sockets Layer de Netscape, que incluye varias aplicaciones, tales como Secure telnet, un módulo para Apache, varias bases de datos como varios algoritmos incluyendo DES, IDEA y Blowfish.

Al usar esta biblioteca, se ha creado un telnet seguro que realiza cifrado sobre las conexiones telnet. A diferencia de SSH, stelnet usa SSL, el protocolo Secure Sockets desarrollado por Netscape. Puede encontrar Secure telnet y Secure FTP empezando con las FAQ SSLeay disponibles en http://www.psy.uq.oz.au/ ftp/Crypto/

PAM - Pluggable Authentication Modules

PAM le permite cambiar sobre la marcha los métodos de verificación, requisitos y encapsular todos los metodos de verificación sin recompilar ninguno de los binarios. La configuración de PAM va más allá de los objetivos de este documento pero asegúrese de echarle un vistazo a la web de PAM para una mayor información http://www.kernel.org/pub/linux/libs/pam/index.html

Sólo algo de lo que puede hacer con PAM:

Usa un cifrado no DES para sus claves. (Haciendo más difícil descifrarlas por la fuerza bruta). Pone límite a los recursos sobre todos los usuarios para que no puedan realizar un ataque de denegación de servicios (número de procesos, cantidad de memoria, etc). Activa shadow passwords sobre la marcha permite a usuarios específicos conectar sólo a horas específicas desde lugares específicos. Con unas pocas horas de instalación y configuración de su sistema puede prevenir muchos ataques antes de que ocurran. Por ejemplo, use PAM para desactivar los archivos rhost del sistema (generales o de punto) en los directorios home de los usuarios añadiendo estas líneas a /etc/pam.d/login:

```
#
# Desactivae rsh/rlogin/rexec para usaurios
#
login auth required pam_rhosts_auth.so no_rhosts
```

Cryptographic IP Encapsulation (CIPE)

El primer objetivo de este software es proporcionar facilidades para asegurar (contra escuchas, incluyendo análisis de tráfico e inyección de mensajes falsos) en comunicaciones entre subredes a través de una red de paquetes inseguros como es Internet.

CIPE cifra los datos a nivel de red. El viaje de los paquetes entre hosts se hace cifrado. La máquina de cifrado está situada cerca del driver que envía y recibe los paquetes.

Esto es, a diferencia de SSH que cifra los datos por conexión, a nivel de socket. Así un conexión lógica entre programas que se ejecutan en hosts diferentes está cifrada.

CIPE se puede usar en tunnelling parrr crear una Red Virtual Privada. El cifrado a bajo nivel tiene la vantaja de poder hacer trabajar la red de forma transparente entre las dos redes conectadas en la RVP sin ningún cambio en el software de aplicación.

Sumario de documentación CIPE :

El estándar IPSEC define un conjuntos de protocolos que se pueden usar (entre otras cosas) para construir RVP. Sin embargo, IPSEC es un protocolo más complicado con un montón de opciones implementadas en todo el protocolo que raramente se usan y algunas características (como gestión de claves) que no están completamente resueltas. CIPE usa una simple aproximación, en la cual muchas cosas que se pueden parametrizar (como la elección del algoritmo actual de cifrado usado) se elijen en el momento de la instalación. Esto limita la flexibilidad, pero permite una implementación simple (y por tanto eficiente, fácil de depurar ...).

Se puede encontrar más información en http://www.inka.de/bigred/devel/cipe.html

Como otras formas de cifrado, esto no se distribuye con el núcleo por defecto debido a restricciones a la exportación.

Kerberos

Kerberos es un método de verificación desarrollado por el Athena Project en el MIT. Cuando un usuario se conecta, la Kerberos verifica que es el usuario (usando clave), y proporciona el usario con una forma de probar su identidad a otros servidores y hosts dispersos en la red.

Esta verificación se usa por programas como rlogin para permitir al usuario conectarse a otros hosts sin clave (en lugar del archivo rhosts). La verificación también se usa por el sistema de correo para garantizar que el correo se entrega a la persona correcta, como también garantizar que el remitente es quien dice ser.

El efecto global de instalar Kerberos y los numerosos programas que van el él, es eliminar virtualmente la posibilidad de que un usuario haga "spoofing"en el sistema, creyendo que es otro. Desafortunadamente, la instalación de Kerberos es muy entrometida, y requiere la modificación o sustitución de numerosos programas estándar.

Puede encontrar más información sobre kerberos en http://www.veritas.com/common/f/97042301.htm y el código se puede encontar en http://nii.isi.edu/info/kerberos/

[From: Stein. Jennifer G., Clifford Neuman, and Jeffrey L. Schiller. "Kerberos: An Authentication Service for Open Network Systems.ÜSENIX Conference Proceedings, Dallas, Texas, Winter 1998.]

Shadow Passwords.

Shadow passwords es un sistema para mantener sus claves cifradas secretas para los usuarios normales. Normalmente estas claves cifradas se almacenan en su archivo /etc/passwd que tiene lectura pública. Esto facilita que alguien pueda ejecutar programas para averiguar las claves en un intento de determinarlas. Shadow passwords guardan sù información en el archivo /etc/shadow que sólo se puede leer con privilegios de superusuario. Para ejecutar shadow passwords tiene que estar seguro de que todas las utilidades que necesitan el acceso a la información de claves están compiladas con para soportar PAM (arriba) también le permite poner un módulo shadow y no requiere la recompilación de ejecutables. Se puede dirigir al Shadow-Password HOWTO para más información si es necesario. Está disponible en http://sunsite.unc.edu/LDP/HOWTO/Shadow-Password-HOWTO.html Está actualizado ahora y no será necesario para distribuciones que soporten PAM.

Crack y John the Ripper

Si por alguna razón su programa passwd no crea claves fuertes le puede interesar ejecutar un programa para romper claves y estar seguro de que las claves son seguras.

Los programas para la rotura de claves funcionan a partir de una idea simple. Intentan todas

las palabras del diccionario y después variaciones sobre estas palabras. Cifra cada un de estas palabras y la compara con la clave cifrada. Si coinciden la hemos encontrado.

Hay varios programas, los más notables son Crackz "John the Ripper"http://www.false.com/security/john/ . Consumen un montón de tiempo de CPU, pero deberían decirle si un atacante podría obtener las claves y después notificárselo a los usuarios con una clave débil. Observe que un atacante primero tendría que usar otro agujero para conseguir las claves cifradas (el archivo unix /etc/passwd), pero esto es más frecuente de lo que podría pensar.

CFS - Cryptographic File System y TCFS - transparent cryptographic File System

CFS es una forma de cifrar un sistema de archivos completo y permitir a los usuarios almacenar archivos cifrados en ellos. Para más información sobre como funciona en: ftp://ftp.research.att.com/dist,

TCFS es una mejora sobre CFS, añadiendo más integración con el sistema de archivos, de forma que es transparente a cualquier usuario que use el sistema de archivos que está cifrado. Más información en: http://edu-gw.dia.unisa.it/tcfs/

X11, SVGA and display security

X11 Es importante asegurar su salida gráfica para prevenir que los atacantes hagan cosas como: grabar sus claves mientras las introduce, sin que lo sepan, leer documentos o información que tiene en la pantalla, o incluso usar un agujero para obtener acceso de superusuario. La ejecución remota de aplicaciones X sobre la red también puede ser llenada con peligro, permitiendo que los espías (sniffers).

X tiene cierto número de mecanismos de control. El más simple es el basado en el host. Puede usar xhost para especificar a qué hosts se les acceder a su display. Esto no es muy seguro de todas formas. Si alguien tiene acceso a su máquina puede hacer xhost + su máquina y entrar fácilmente. También, si tiene que permitir acceso de una máquina insegura cualquiera puede comprometer su display.

Cuando use (x display manager) para entrar al sistema, tiene un método mucho mejor de acceso: MIT-MAGIC-COOKIE-1. Se genera un cookie de 128bit y se almacena en su archivo .Xauthorty. Si necesita permitir acceder a su display a una máquina remota, puede usar el el comando xauth y la información de archivo .Xauthority para proporcionar acceso a esa conexión. Vea el mini-howto Remote-X-Apps disponible en http://sunsite.unc.edu/LDP/HOWTO/mini/Remote-X-Apps.html.

También puede usar ssh (vea ssh arriba) para permitir conexiones X seguras. Esto tiene la ventaja de ser también transparente al usuario final, y significa que no circulan por la red datos descifrados.

Mire la página de manual de Xsecurity para mas información sobre la seguridad en X. La mejor apuesta es usar xdm para conectarse a su consola y entonces ssh para ir a los sitios remotos en los que quiere ejecutar aplicaciones.

SVGA Los programas de SVGAlib son típicamente SUID-root para poder acceder a todo su hardware de video de su máquina Linux. Esto los hace muy peligrosos. Si fallan, típicamente necesita reiniciar su máquina para tener una consola utilizable. Esté seguro de que cualquier programa SVGA que ejecuta es auténtico, y que al menos se puede confiar. Incluso mejor, no los ejecute.

GGI (Generic Graphics Interface project) El proyecto Linux GGI está intentando resolver varios problemas con los interfaces de video sobre Linux. GGI moverá una pequeña parte del código de video al núcleo de Linux y entonces controlar el acceso al sistema de video. Esto significa que GGI podrá restaurar su consola en cualquier momento a un estado bueno conocido. También permitirá una clave de atención segura para que pueda estar seguro que no hay ningún caballo de Troya de login ejecutándose en su consola. http://synergy.caltech.edu/ ggi/

Seguridad del Núcleo

Esto es una descripción de las opciones de configuración del núcleo que están relacionadas con la seguridad y una explicación de lo que hace y como usarlo.

Como el núcleo controla la red de su equipo, es importante que el núcleo sea muy seguro, y que el mismo núcleo no pueda ser comprometido. Para prevenir algunos de los últimos ataques de red, debe intentar mantener una versión del núcleo actualizada. Puede encontrar los nuevos núcleos en ftp://ftp.kernel.org.

Opciones de Compilación del Núcleo

IP: Drop source routed frames (CONFIG_IP_NOSR) Esta opción debería estar activada. Source routed frames contienen la ruta completa de sus destinos dentro del paquete. Esto significa que los enrutadores a través de los que circula el paquete no necesitan inspeccionarlo, y sólo lo reenvían. Esto podría llevar a que los datos que entran a su sistema puedan ser un exploit potencial.

IP: Firewalling (CONFIG_IP_FIREWALL) Esta opción es necesaria si va a configurar su máquina como un cortafuegos, hacer enmascaramiento o desea proteger su estación de trabajo con línea telefónica de qeu alguien entre a través de su interfaz PPP.

IP: forwarding/gatewaying (CONFIG_IP_FORWARD) Si activa reenvío IP (IP forwarding), su caja Linux esencialmente se convierte en un encaminador (router). Si su máquina está en una red, podría estar enviando datos de una red a otra, y quizás subvirtiendo un cortafuegos que esté puesto allí para evitar que esto suceda. Los usuario normales de conexión telefónica les interesará desactivar esto y otros usuarios se deberían concentrar en las implicaciones de seguridad de hacer esto. Las máquinas cortafuegos querrán esto activada y usarlo en conjunción con el software de cortafuegos.

Puede activarr y desactivar el reenvío IP (IP forwarding) dinámicamente usando el siguiente comando:

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

y desactivarlo con el comando:

```
root#  echo 0 > /proc/sys/net/ipv4/ip_forward
```

Ese archivo (y muchos otros archivos de /proc) aparecerán con longitud cero, pero de echo no lo son. Esto es una nueva característica introducida, por lo que tiene que tener un núcleo 2.0.33 o posterior.

IP: firewall packet logging (CONFIG_IP_FIREWALL_VERBOSE) Esta opción le da información sobre los paquetes que su cortafuegos recibe, como remitente, recipiente, puerto, etc.

IP: always defragment (CONFIG_IP_ALWAYS_DEFRAG) Generalmente esta opción está desactivada, pero si esta construyendo un host cortafuegos o apra enmascaramiento, deberá activarla. Cuando se envía de un host a otro, no siempre se envía como un simple paquete de datos, sino que se fragmenta en varios trozos. El problema de esto es que los números de puerto sólo se almacenan en el primer fragmento. Esto significa que alguien puede insertar información en el resto de los paquetes para su conexión que se supone que no deberían estar allí.

IP: syn cookies (CONFIG_SYN_COOKIES) El ataque SYN es un ataque de denegación de servicio (denial of service DoS) que cosume todos los recuroso de su máquina forzando un reinicio. No podemos encontrar ninguna razón por la que no debiera activar esto.

Packet Signatures (CONFIG_NCPFS_PACKET_SIGNING) Esta es ua opción disponible en los núcleos de la serie 2.1 que firmarán los paquetes NCP para una mayor seguridad. Normalmente puede dejarlo desactivado,-pero está allí por si lo necesita.

IP: Firewall packet netlink device (CONFIG_IP_FIREWALL_NETLINK) Esta es un hábil opción que le permite analizar los primeros 128 bytes de los paquetes en el espacio de programa de usuario, para determinar si le gustaría aceptar o denegar el paquete basado en su validez.

Dispositivos del Núcleo

Hay algunos dispositivos de bloque y carácter disponibles en Linux que también ayudarán con la seguridad.

Los dos dispositivos /dev/random y /dev/urandom los proporciona el núcleo para recuperarar datos aleatorios en cualquier instante.

Ambos, /dev/random y /dev/urandom deberían ser suficientemente seguros como para generar claves PGP, SSH y otras aplicaciones donde son un equisito números aleatorios seguros. Los atacantes no podrían ser capaces de determinar el siguiente número dada cualquier secuencia de números con este origen. Se han puesto muchos esfuerzos para asegurar que los números que obtiene de esta fuente son aleatorios en todos los sentidos de la palabra aleatorio.

La única diferencia es que /dev/random suministra bytes aleatorios y le hace esperar para que se acumulen más. Observe que en algunos sistemas puede bloquear durante un rato a la espera de que se genere una nueva entrada de usuario al sistema. Por tanto debe tener cuidado al usar /dev/random. (Quizás lo mejor que puede hacer es usarlo cuando esté generando información sensible de claves e indicarle al usuario que pulse una tecla repetidas veces hasta que indique por la pantalla .ºK, es suficiente").

/dev/random tiene gran calidad e entropía, midiendo tiempos entre interrupciones etc. Bloquea hasta que hay dispoibles suficientes bits de datos aleatorios.

/dev/urandom es parecido, no es tan seguro, pero suficiente para la mayoría de las aplicaciones.

Puede leer los dispositivos usando algo parecido a lo siguiente:

```
root# head -c 6 /dev/urandom | uuencode -
```

Esto imprimirá seis caracteres aleatorios en la consola, válidos para la generación de una

clave. Vea /usr/src/linux/drivers/char/random.c para obtener una descripción del algoritmo.

Seguridad de Red

La seguridad en redes se está volviendo más y más importante ya que la gente pasa cada vez más tiempo conectada. Comprometer la seguridad de una red es con frecuencia más fácil que hacerlo con la física o local, y mucho más frecuente.

Hay un gran número de herramientas para ayudar con la seguridad de una red y mucha información que viene con las distribuciones de Linux.

Espías de paquetes (Packet Sniffers)

Una de las formas más frecuentes que tienen los intrusos para obtener acceso a más sistemas de su red es usar un espía de paquetes (sniffer) en un host que ya ha sido comprometido. Este "sniffer"escucha en los puertos Ethernet cosas como "Password", "Login"su"en el flujo de paquetes y registra el tráfico posterior. De esta forma los atacantes obtienen claves de sistemas que ni siquiera tratan de atacar. Las claves enviadas sin cifrar son muy vulnerables a estos ataques.

EJEMPLO: El host A ha sido comprometido. Los atacantes instalan un sniffer. El sniffer recoge los logind del administrador en el host B desde el host C. Consigue las claves personales de los administradores mientras entran en B, entonces el administrador hace un su para corregir un problema. Ahora tienen la clave del root del host B. Más tarde el administrador deja a alguien hacer telnet desde su cuenta al host Z en otro sistio. Ahora el atacante tiene el par password/login en el host Z.

En este momento el atacante no necesita ni comprometer el sistema para hacer esto, también podría traer un portátil o un pc al edificio y meterse en su red.

El uso de ssh u otros métodos de claves cifradas impide este ataque. Cosas como APOP para cuentas pop también previene este ataque. (Los login pop normales son muy vulnerables en esto, como cualquier cosa que se envía en texto sin cifrar sobre la red).

Verificar su Información DNS

Mantener actualizada la información DNS sobre todos los hosts de su red le puede ayudar a aumentar la seguridad. En el caso de un host no autorizado se conecte a su red, puede reconocerlo por la ausencia de la entrada DNS. Muchos servicios se pueden configurar para no aceptar conexiones de hosts que carecen de entradas DNS válidas.

identd

identd es un pequño programa que se ejecuta desde su inetd. Mantiene la pista de qué usuario está ejecutando un determinado servicio tcp, e informa de ello cuando se le solicita.

Mucha gente confunde la utilidad de identd, y en consecuencia lo desactiva o bloquea todos los sitios que lo solicitan. Identd no está ahí para ayudar a sitios remotos. No hay forma de conocer si los datos que que obtiene del identd remoto son correctos o no. No hay verificación de identidad en las solicitudes de identd.

Enonces ¿por qué querría ejecutarlo? Porque le ayuda y es otra fuente de información. Si su identd no está alterado, entonces sabe que le está diciendo el usuario o uid de la gente de los sitios remotos que están usando los servicios tcp. Si el administrador de un sitio remoto viene y le dice que un usuario está intentando entrar (hack) en su sitio, entonces puede fácilmente tomar acciones contra ese usuario. Si no está ejecutando identd, tendrá que mirar un montón de registros para encontrar quien fue a determinada hora, y en general lleva mucho más tiempo encontrar la pista del usuario.

El identd que viene con la mayoría de las distribuciones es más configurable de lo que mucha gente cree. Puede desactivar identd para determinados usuarios (pueden crear un archivo .identd), puede registrar todas las peticiones de identd (lo recomiendo) puede incluso tener a identd devolviendo un uid en lugar del nombre de usuario o incluso NO-USER.

SATAN , ISS y otros Exploradores de Red

Hay varios paquetes diferentes de software que efectúan una exploración basada en puertos y servicios de máquinas o de redes. SATAN y ISS son dos de los mejores conocidos. Este software se conecta con la máquina destino (o todas las máquinas de la red) en todos su puertos que puede e intenta determinar qué servicio se está ejecutando allí. Basándose en

esta información, podría descubrir si la máquina es vulnerable a un exploit específico en ese servidor.

SATAN (Security Administrators Tool for Analyzing Networks) es un explorador de puertos con un interfaz web. Se puede configurar para efectuar verificaciones ligeras, medias o fuertes sobre una máquina o red, y fija los problemas que encuentra. Esté seguro de obtener una copia de SATAN de sunsite o un FTP o web con reputación. Había un copia Troyano de SATAN que se distribuía por la red. http://www.trouble.org/ zen/satan/satan.html

ISS (Internet Security Scanner) hay otro explorador basado en puertos . Es más rápido que Satan y en consecuencia podría ser mejor para grandes redes. Sin embargo, SATAN tiende a proporcionar más información.

Abacus-Sentry es un explorador de puertos comercial de www.psionic.com. Mire su página home en la web para más información. http://www.psionic.com/

Detecting Port scans. Hay algunas herramientas diseñadas para alertarle de pruebas con Satan, ISS y otro software de exploración. De todas formas el uso de tcp_wrappers y estando seguro de buscar en los archivos regulares de registro debería notar tales pruebas.

Sendmail, qmail y MTA's.

Uno de los servicios más importantes que puede proporcionar es el servidor de correo. Por desgracia, también es uno de los servicios más vulnerable a los ataques, simplemente debido al número de tareas que debe realizar y los privilegios que típicamente necesita.

Si usa sendmail es muy importante mantener la versión actualizada Sendmail tiene una larga, larga historia de fallos de seguridad. Siempre esté seguro de usar la versión más reciente. http://www.sendmail.org/

Si está cansado de actualizaciones de su versión de sendmail cada semana debería plantearse cambiar a qmail. Qmail se diseñó teniendo en mente la seguridad desde principio a fin. Es más rápido, estable y seguro. http://www.qmail.org/

Denegación de Servicios

Un ataque de denegación de serviciso es uno en que el atacante intenta hacer que algún recurso esté demasiado ocupado para responder solicitudes legítimas o para denegar a los usuarios legítimos acceso a su máquina.

Los ataques de denegación de servicios se han incrementado recientemente en los últimos años. Algunos de los más populares y recientes están listados abajo. Observe que aparecen nuevos continuamente, por tanto esto son sólo algunos ejemplos. Lea la lista de seguridad Linux y la lista bugtraq y archivos para una información más actualizada.

SYN Flooding - SYN flooding es un ataque de denegación de servicio de red. Se aprovecha de un agujero ("loophole") en la forma en que se crean las conexiones TCP. Los nuevos núcleos de Linux (2.0.30 y posteriores) tienen varias opciones configurables para prevenir ataques SYN flood que denieguen a la gente acceder a su máquina o servicios. Vea la sección de seguridad del núcleo para ver las opciones limpias de rotección del núcleo.

Pentium "F00F"Bug - Recientemente se descubrió que una serie de código ensamblador enviado a un Pentium genuino reiniciabala máquina. Esto afecta a todas las máquinas con un procesador Pentium (no clones, no Pentium Pro o PII), no importa qué operación de sistema esté realizando. Los núcleos Linux 2.0.32 y superiores tienen un trabajo sobre este bug, previniendo que bloquee la máquina. El múcleo 2.0.33 tiene una versión mejorada de la corrección, sugerida sobre 2.0.32. Si está ejecutando un Pentium debería actualizarse ahora.

Ping Flooding - Ping flooding es un simple ataque de denegación de servicio por la fuerza bruta. El atacante envía una "inundación"(flood) de paquetes ICMP a su máquina. Si están haciendo esto desde un host con mayor ancho de banda que su máquina, será incapaz de enviar nada a la red. Una variante de esta ataque llamado "smurfing", envía paquetes ICMP a un host con la dirección IP de retorno de su máquina, permitiéndoles que la inundación sea menos detectable. Puede encontrar más información sobre ataques "smurf"en http://www.quadrunner.com/ chuegen/smurf.txt Si en alguna ocasión está bajo un ataque ping flood, use una herramienta como tcpdump para determinar de donde vienen los paquetes (o parece que vienen), entonces contacte con su proveedor con esta información. Los ping floods se pueden parar con más facilidad a nivel de encaminador o usando un cortafuegos.

Ping o' Death - El ataque Ping o' Death es el resultado de paquetes entrantes ICMP ECHO REQUEST que son más grandes que lo que pueden almacenar las estructuras

de datos del núcleo que recogen esta información. Como enviar un simple paquete ping grande (65,510 bytes) "ping.ª muchos sistemas, los cuelga o los rompe, este problema fue rápidamente resuelto. Este ha sido resuelto y ya no hay nada de qué preocuparse.

Teardrop / New Tear - Uno de los más recientes exploits que usan un bug presente en el código de fragmentación IP en plataformas Linux y Windows. Se corrigió en la versión del núcleo 2.0.33, y no requiere seleccionar ninguna opción de compilación del núcleo para usar la corrección. Linux aparentemente no es vulnerable al exploit 'newtear'.

Puede encontrar más código de exploits y más descripciones en profundidad de como funcionan en http://www.rootshell.com/ usando su motor de búsqeda.

Seguridad NFS (Network File System)

NFS es un protocolo para compartir archivos ampliamente utilizado. Permite a los servidores ejecutar nfsd y mountd para "exportar"sistemas de archivos completos a otras máquinas con soporte de sistemas de archivos nfs incorporado en el núcleo (u otro cliente que lo soporte si no son máquinas Linux). Mountd mantiene una pista de los sistemas de archivos montados en /etc/mtab, y puede mostrarlos con 'showmount'.

Muchos stios usan NFS para servir directorios home a sus usuarios, de forma que sin importar en qué máquina del grupo se conectan, tienen su directorio home y archivos.

Hay alguna parte de "seguridad "permitida al exportar sistemas de archivos. Puede hacer que su nfsd mapee al usuario root remoto (uid=0) al usuario nobody, denegándole acceso total a los archivos exportados. Sin embargo, como los usuario individuales tienen acceso sus propios archivos (o al menos con el mismo uid), el superusuario remoto puede entrar en su cuenta y tener acceso total a sus archivos. Esto es sólo un pequeño inconveniente para un atacante que tiene acceso para montar su sistema de archivos remoto.

Si tiene que usar NFS, esté seguro de que exporta sólo a aquellas máquinas que realmente necesita exportar. Nunca exporte su directorio raíz completo, exporte sólo los directorios que necesita exportar.

NIS (Network Information Service) (conocido como YP).

Network Information service (llamado YP) es un medio de distribuir información a un grupo de máquinas. El NIS principal recoge las tablas de información y las convierte en archivos de

mapas NIS. Estos mapas son servidos sobre la red, permitiendo a las máquinas NIS clientes conseguir login, claves, directorio home e información de shell (toda la información estándar del archivo /etc/passwd file). Esto permite a los usuarios cambiar sus claves una vez y que tenga efecto en todas las máquinas del dominio NIS.

NIS no es del todo seguro. No nació para esto. Nació para ser útil y manejable. Cualquiera que puede averiguar el nombre de su dominio NIS (cualquiera de la red) puede obtener una copia del archivo passwd, y usar crack o john the ripper con las claves de usuarios. También es posible falsificar NIS y hacer toda clase de trucos sucios. Si tiene que uar NIS, esté seguro de que está prevenido de los peligros.

Hay una sustitución de NIS mucho mas segura, llamada NIS+. Mire el NIS HOWTO para más información: http://sunsite.unc.edu/mdw/HOWTO/NIS-HOWTO.html

Esto significa que cada vez que se añade un nuevo RPM al sistema, la base de datos RPM tendrá que ser rearchivada. Tendrá que decidir entre ventajas e inconvenientes.

Actualizaciones del Sistema

La mayoría de los usuarios de Unix instalan desde un CDROM. Debido a la rapidez con que corrigen los problemas de seguridad, siempre están apareciendo nuevas versiones de los programas. Antes de que conecte su máquina a la red es una buena idea verificar con el sitio ftp se su distribución (ftp.redhat.com por ejemplo) y conseguir todas los paquetes de actualizaciones desde que salió su CDROM de distribución. Muchas veces estos paquetes contienen importantes correcciones de seguridad, por lo que es conveniente tenerlos instalados.

¿Qué hacer Durante y Después de una Ruptura?

Lo primero es mantener la calma. Las acciones apresuradas pueden causar más daño que el que podría causar un atacante.

Si ha detectado a un usuario local intentando comprometer la seguridad lo primero que tiene que hacer es confirmar que realmente es quien parece ser. Compruebe el sitio si tiene registros de login. ¿Está en el sitio habitual? Entonces use un medio no electrónico para ponerse en contacto. Por ejemplo, llámelo por teléfono o vaya andando por el edificio/casa y hable con ellos/ellas. Si reconocen que están conectados, puede preguntarles la explicación qué están haciendo o decirles que dejen de hacerlo. Si no están conectados y no tienen ni idea de lo que

les está hablando, este incidente requiere una mayor investigación. Busque en el incidente, y consiga un montón de información antes de hacer acusaciones.

Si ha detectado un compromiso de red, lo primero que tiene que hacer (si puede) es desconectar la red. Si están conectados mediante módem, desconecte el cable del módem. Si están conectados vía ethernet, desconecte el cable de ethernet. Esto prevendrá que hagan más daño, y ellos además lo verán más como un problema de red más que una detección.

Si no puede desconectar la red (si tiene un sitio ocupado, o no tiene control físico de la máquina), el mejor paso siguiente es usar algo como tcp_wrappers o ipfwadm para denegar accesos del sitio de donde se hace la intrusión.

Si no puede denegar a toda la gente del mismo sitio del intruso, cierre la cuenta del usuario. Observe que cerrar una cuenta no es una cosa simple. Tiene que tener en cuenta los archivos .rhosts, el acceso FTP y otras puertas traseras.

Tras haber hecho lo anterior (desconectar la red, denegar el acceso de sitio y/o desactiva la cuenta), necesita matar todos sus procesos de usuario y desconectarlos.

Debería monitorizar su sitio bien durante los próximos minutos, ya queel atacante intentará regresar. Quizás usando un cuenta diferente y/o de diferente dirección de red.

Cerrar el Agujero Si puede determinar qué medios usó el atacante para entrar en su sistema, debería intentar cerrar el agujero. Por ejemplo, quizás ha visto varias entradas FTP justo antes de que el usuario entrara. Desactive el servicio FTP y compruebe y vea si hay una versión actualizada o alguna de la lista conocida de correcciones.

Verifique todos su archivos de registro y haga una visita a sus páginas de listas de seguridad y vea allí su hay algún nuevo exploit común que pueda corregir. Es muy probable que alguno delos vendedores haya editado una actualización de seguridad, que la mayoría de los otros Unix también harán.

Si no elimina el atacante, probablemente volverá. No a su máquina, sino a cualquiera otra de la red. Si ejecutaban algún sniffer, tiene buenos recursos para tener acceso a las máquinas locales.

Valoración del Daño Lo siguiente que tiene que hacer es evaluar el daño. ¿Qué se ha comprometido? Si ejecuta un Verificador de Integridad como Tripwire podría decírselo. Si no, tendrá que mirar todos sus datos importantes.

Como los sistemas Linux se están volviendo más y más fáciles de instalar, podría considerar guardar sus archivos de configuración y limpiar los discos y reinstalar, y después restaurar sus archivos de usuario de las copias de seguridad y los archivos de configuración. Esto asegura que tiene un sistema limpio. Si tiene archivos de copias de seguridad del sistema comprometido, sea especialmente cuidadoso con los binarios que restaura, ya que pueden ser caballos de troya situados allí por el intruso.

Backups Si su sistema está comprometido, puede restaurar los datos que necesite de estas copias. Desde luego, algunos datos tienen valor para los atacantes y no sólo los destruirán , los robarán y tendrán sus propias copias, pero al menos tiene los datos. Debería verificar varias copias de salvaguardia anteriores antes de restaurar un archivo que haya sido falsificado. El intruso podría haber comprometido sus archivos hace tiempo y podría haber hecho muchas copias de seguridad del archivo falsificado.

Desde luego, también tiene que tener seguras sus copias de seguridad. Tenga cuidado de guardarlas en lugar seguro. Saber quien tiene acceso a ellos. (Si un atacante puede obtener las copias de seguridad, puede obtener acceso a todos los datos sin que ni siquiera lo sepa).

Tracking Down the Intruder. Ha expulsado al intruso y ha recuperado su sistema, pero no todo está hecho. Mientras sea improbable que la mayoría de los intrusos sean capturados, debería informar del ataque.

Debería informar del ataque al contacto con el admin en el sitio de donde el atacante atacó a su sistema. Puede buscar este contacto con whois o la base de datos del internic. Podría enviarles un mensaje de correo con todos los registros aplicables y fechas y horas. Si tiene algo más distintivo sobre su intruso, podría mencionarlo también. Tras enviar el correo (si le parece bien) podría llamar por teléfono. Si el admin localiza a su atacante, podría hablar con él.

Los buenos hackers con frecuencia usan muchos sistemas intermedios. Algunos (o muchos) puede que ni sepan que han sido comprometidos. Intentar seguir la pista de un cracker hasta su origen puede ser difícil. Siendo educado con los admins le puede llevar un largo recorrido obtener ayuda de ellos.

Debería notificarlo también a alguna organización de seguridad de la que foram parte (CERT o similar).

Security Sources Es muy importante suscribirse a una o más listas de correo de seguridad y estar actualizado sobre las correcciones de seguridad. La mayoría de las listas tienen muy bajo volumen y son muy informativas.

4.5.3. Instalación del servidor

El sistema operativo va ser dependiente de las necesidades del recursos como mplementación que se van a correr, sin menosprecias las políticas que se implanten. Por ejemplo si va ha manejar una base de datos, y la información es generada por un servidor de arquitectura risk Silicon Graphics y la información del equipo es de grandes capacidades para nuestra red lo conveniente es dejar la información ene sitio con su base de datos, pero se debe de establecer como una maquina de alto riesgo por su alto índice de agujeros de seguridad y no colocarla como un servicio de información publica como ftp o http, amenos de que este con múltiples implementaciones de seguridad para minimizar los riesgos. se ha empezado de hablar de los índices de vulnerabilidad, estos los pueden actualizados en su distribuidor de sistema operativo o software o hardware. Con respecto a los ya implicados o comprometidos en ataques previos. Retomando el ejemplo si usted tiene un servidor Silicon Graphics, tendrá como sistema operativo nativo Irix, y dependiendo del tipo de arquitectura su correspondiere versión de sistema operativo por ejemplo Irix 6.5 con su actualización 6.5.X.X y la información de seguridad la encuentra en `http://www.sgi.com/security`. No se puede generalizar el tipo de instalación debido a la arquitectura, el sistema operativo, los recursos de la empresa como la disposición del equipo de computo. así que a grandes rasgos y sin introducirme mucho al tema, daré recomendaciones del tipo de instalación.

Sistema de archivos

La manera de partiocionar el disco duro dependerá del espacio de nuestros recursos y el motivo del servidor. Tenemos como un máximo de 7 particiones por disco duro en un servidor, bajo las recomendaciones de Unix V. Al menos Unix propone montar una sección `raíz /` donde se montaran los archivos del administrador. Otra partición es la denominada partición `SWAP` o memoria virtual ubicada en el disco duro, esta memoria deberá ser el doble de la memoria física Se recomienda si va alojar usuarios asignarlos en una partición distinta ya se `/home` o `/usr/people` o `/usr/export/home` y va a depender del tamaño de los usuarios y su rendimiento con el sistema, Una utilería que podemos colocar para la administración de espacio es cuota la cual limita al usuario a un espacio máximo. Otra partición importante en nuestro sistema es el uso de binarios o programas de ecuación que se ubican en /usr, existe una ventaja en el uso de aplicaciones como el acceder la información de solo lectura, se refiere a que el ejecutable solo puede ser leído mas no modificado por cualquier usuario, la diferencia de permisos a acceso a una partición es que para que un atacante desee modificar a través de un *rootkit*, tendrá que montar el sistema de archivos mas la modificación esto es mas tiempo de demora para su ingreso o modificación del sistema.

Programas que se van a instalar

Dependiendo de las necesidades de cada aplicación va ha depender

Hay varias distribuciones de Linux, y entre sí pueden ser tan diferentes como dos Unix es sin relación entre sí. Afortunadamente, han mostrado una tendencia a converger sobre un número mucho menor de familias:

Redhat

Casi todas las distribuciones hoy en día están basadas en RedHat [1] - LinuxPPP [2], Conectiva [3], Hispafuentes [4], Mandrake [5], TurboLinux [6], y en cierta medida SuSE [7] y Caldera [8] juegan con las mismas reglas. Todos estos sistemas se centran en una instalación sencilla; todos ellos ofrecen ayuda gráfica para la configuración del sistema por medio de linuxconf, drakconf, yast o algún equivalente. La selección de paquetes se puede hacer por categorías o por paquetes individuales. Hay estilos de configuración pre-establecidos, reduciendo el proceso de instalación a un par de teclazos - o más aún, de clicks de mouse. RedHat y sus derivados son, y por mucho, líderes en el mercado. Últimamente se han orientado cada vez más a usuarios novatos, descuidando importantes aspectos de seguridad a favor de facilidad de uso. Mucha gente indica como recomendable para un usuario nuevo iniciar con una de estas distribuciones, pues son para las que más fácilmente encontrará soporte. Manejan el sistema de paquetes RPM (RedHat Package Manager), que evita que se rompan dependencias y es un muy buen auxiliar para mantener al sistema en un estado consistente. Con una cantidad razonable de esfuerzo, es posible cerrar las principales vulnerabilidades de una de estas distribuciones, logrando una instalación mucho más segura que la que tienen de fábrica.

Debián

Debian [9] nace como un proyecto comunitario, fuertemente basado en la ideología de la Free Software Foundation [10]. Tienen un fuerte contrato social [11], en el que se comprometen con la comunidad de software libre, y en el cual está basada la Open Source Definition [12], escrita originalmente por Bruce Perens. Debian es desarrollado y mantenido por una comunidad, no por una empresa. Esto hace que no tenga presiones para sacar versiones; los sistemas Debian son realmente robustos, los paquetes que aparecen en su rama estable han sido ampliamente probados; el sistema de paquetes de Debian (.deb, manejado con dpkg y apt-get, y varios front-ends como dselect y aptitude) es el más avanzado en el mundo de Linux; actualizar un sistema Debian completo puede hacerse con sólamente dos líneas de apt-get; la resolución de dependencias y conflictos es muy superior inclusive a la de RPM. sin embargo, en aras de la madurez técnica, se ha sacrificado el lado de la interfaz de usuario, dando como resultado

un sistema que a entender de muchos no es muy apto para principiantes.

Slackwaretas

Cada día menos comunes, pero con una base de leales usuarios. Slackware [13] es la más veterana de las distribuciones originales de Linux que se siguen manteniendo - SLS e Yggdrasil, los verdaderos pioneros, desaparecieron tras cierto tiempo. Slackware se mantiene fiel a su clientela, que busca un Unix tradicional. Si no me equivoco Slackware es la única distribución de Linux hoy en día que utiliza un sistema de arranque tipo BSD [14] (contrastando con el arranque tipo SysVR4). Slackware no utiliza un sistema de paquetes como los .deb o los .rpm, sino que paquetes .tgz al estilo de los Unix tradicionales, con un muy débil manejo de versiones, por lo que actualizar normalmente implica recompilar, y desinstalar un paquete puede desencadenar una desagradable cadena de dependencias no resueltas - O peor aún, al intentar actualizar algo puedes, sin darte cuenta, terminar instalando versiones previas. Sin embargo, bien administrado tiene fama de ser muy robusto y estable. Slackware cuenta con una base de usuarios muy limitada, y generalmente son usuarios relativamente experto.

Minimalistas

En Linux existe una gran cantidad de minidistribuciones, conjuntos pequeños de programas con un propósito específico hechos para computadoras de propósito específico (p. ej. ruteadores o firewalls), máquinas con prestaciones muy reducidas, o para revivir sistemas dañados. Algunas de las más populares son muLinux [15] (distribución en varios floppies que busca dar lo más cercano a una distribución completa de Linux en floppies y sin necesidad de disco duro), floppyFW [16] (Un sencillo ruteador con capacidades básicas de firewall en un sólo floppie), Linux Router Project [17] (Otro proyecto que busca crear un ruteador. Más extensible que floppyFW), Trinux [18] (Juego extensible de herramientas de seguridad en floppies), Hal91 [19] (Distribución minimalista, muy utilizada para construir aplicaciones específicas sobre de ella), LODS [20] (Derivado de Hal91, incluye un VNCViewer sobre svgalib para tener siempre a la mano una estación de manejo remoto gráfico) y tomsrtbt [21] (Las herramientas básicas para la recuperación de un sistema dañado).

4.6. Una instalación segura es instalar solo los recursos necesarios

Es práctica muy recomendable el no conectar la computadora a red recién terminada la instalación, sino que sólo hacerlo tras haberla asegurado.

4.7. RedHat

Es muy diferente instalar una estación de trabajo e instalar un servidor. RedHat ofrece varias opciones predeterminadas de instalación rápida: Estación de trabajo GNOME, estación de trabajo KDE, servidor, o personalizada. Se sugiere fuertemente nunca elegir una opción predeterminada. Hacerlo equivale a solicitarle que instale todo lo que quien preparó la distribución cree que puede valer la pena – tanto para servidor como para estación de trabajo, es altamente recomendable seleccionar instalación personalizada.

De hecho, si bien tomará mucho más tiempo, al instalar un servidor –una computadora que será visible a la red externa, que prestará servicios y probablemente sea blanco de ataques– te recomiendo ampliamente seleccionar las categorías de paquetes que requieras, y después de eso indicar al instalador que quieres seleccionar paquete por paquete qué instalar.

4.8. Debian

4.8.1. Cerrar puertos de servicio

Hay varios protocolos que presentan un riesgo adicional: Por más seguro que sea un demonio, si requiere que la contraseña sea transmitida en texto claro (como lo hacen FTP, telnet, POP3, IMAP, etc.) abre nuestras máquinas a que un atacante olfatee la red esperando encontrar una contraseña, y la utilice para suplantar la identidad de su dueño.

Una herramienta invaluable para asistirte al controlar acceso a los servicios que proporcionamos en nuestro sistema es un firewall, ya sea como un sistema dedicado o como reglas locales de filtrado, como lo veremos en .

inetd /xinetd

Vayamos primero sobre del superdemonio, inetd. Este superdemonio está encargado de levantar una potencialmente gran cantidad de programas servidor para varios servicios, típicamente de baja complejidad pues funciona mejor con programas que no cueste mucho tiempo inicializar. Bajo el riesgo de equivocarme en alguno, en inetd vienen abiertos varios servicios, como chargen, echo, telnet, ftp, rsh, rexec, rwho, talk, finger, ident. Además de ser innecesar-

ios hoy en día, pueden ser muy peligrosos. ¿Ejemplos? Con identd, finger, rwho y chargen es posible averiguar información acerca de quién está usando y qué está corriendo una computadora –información, claro, muy valiosa para un atacante. Varios de estos demoniostambién han presentado casos de buffer overflows.

El archivo de configuración de inetd es el /etc/inetd.conf, y su sintaxis es muy sencilla.

Todas las líneas que no son comentarios y no están en blanco inician con ya sea una palabra o un número. Éste es el que define qué servicio ejecutará – por ejemplo, la línea:

```
telnet stream tcp nowait telnetd.telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd
```

Indica que estamos hablando de telnet. Claro, la computadora tendrá que traducir esto a un número de puerto; para esto extá el archivo /etc/services, donde nos indica que telnet corresponde al puerto 23 de TCP. Si queremos deshabilitar ese servicio, únicamente tenemos que comentar la línea en cuestión, de esta manera:

#telnet stream tcp nowait telnetd.telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd

Después de hacer los cambios pertinentes, reiniciamos inetd. Esto puede ser -dependiendo del tipo de Unix que tengamos- de una de las siguientes maneras:

/etc/init.d/inetd stop; /etc/init.d/inetd start

/etc/rc.d/init.d/inetd stop;/etc/rc.d/init.d/inetd start

Cabe mencionar que la algunas versión, incluye a un reemplazo para inetd llamado xinetd. La configuración general, localizada en el archivo /etc/xinetd.conf es:

```
defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}

includedir /etc/xinetd.d
```

Con xinetd tenemos un control más granular acerca del comportamiento de cada uno de los servicios. Por ejemplo, en la configuración anterior vemos que xinetd limitará a 60 instancias (conexiones simultáneas), lo cual puede reducir dramáticamente los ataques de negación de servicio. En este ejemplo vemos también cómo las conexiones serán registradas en las bitácoras del sistema. Y en vez de definir cada servicio en una línea, en el directorio /etc/xinetd.d tenemos un archivo por servicio (toman el nombre de las líneas definidas en /etc/services) como el siguiente, /etc/xinetd.d/telnet:

```
service telnet
{
        flags           = REUSE
        socket\type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        logon\failure  += USERID
        disable         = yes
}
```

La mayoría de los campos son análogos a los que vemos en inetd.conf – La única diferencia clara es el campo enable. En vez de comentar la línea, basta con marcar disable=yes para que el servicio no sea iniciado.

4.9. Demonios independientes

Por otro lado, tenemos los demonios por sé, los programas que están contínuamente corriendo en nuestro sistema. En un sistema Linux típico basado en SysVR4 (no Slackware), encontraremos qué programas demonio tenemos corriendo examinando los directorios de inicio. Para esto, consultamos en qué runlevel estamos ejecutando (busca la línea id:x:initdefault: en tu archivo /etc/inittab; el número que esté en la x será el runlevel en el que entra tu sistema por default.

Dentro de /etc o de /etc/rc.d, encontrarás los directorios init.d, rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d y rc6.d. Algunos sistemas tendrán también a rcS.d. En init.d están los scripts de arranque/finalización de todos tus demonios, y en los demás directorios (uno por cada runlevel) hay ligas simbólicas hacia dichos scripts, con una convención especial en el nombre:

```
{K|S}##xxxx
```

, donde:

K Terminar el demonio al entrar al runlevel indicado S Iniciar el demonio al entrar al runlevel indicado ## El órden en el que será iniciado/terminado (ascendente) xxxx Nombre del demonio

Por ejemplo, si entramos en runlevel 2 por default y tenemos los siguientes archivos en /etc/rc2.d:

```
gwolf@mipc:/etc\$ ls rc2.d/
S10sysklogd  S20dhcp  S20inetd   S20lpd         S20snort  S89cron      S99wdm
S12kerneld   S20exim  S20ipac    S20makedev     S20ssh    S91apache
S14ppp       S20gpm   S20logoutd S20postgresql  S89atd    S99rmnologin
```

Esto significa que al encender el sistema o entrar a runlevel 2, éste inicia (en órden) la ejecución de sysklogd, kerneld, ppp, dhcp, exim, gpm, inetd, ipac, logoutd, lpd, makedev, postgresql, snort, ssh, atd, cron, apache y rmnologin.

Para evitar que un demonio inicie en el runlevel default basta con quitar la liga hacia él; por ejemplo, si decides que ya no requieres dar servicios de Apache, basta con la siguiente línea: texttrm -f /etc/rc2.d/S91apache

O bien, si usas Debian, puedes hacer:

```
update-rc.d -f apache remove
```

El crear la liga con la K nos sirve para indicar que al entrar a este runlevel, en caso de que cierto demonio esté corriendo, lo mate. Recuerda que podemos cambiar de runlevels sin reiniciar el sistema, con el comando

init x

Siendo x el runlevel al que queremos entrar.

4.10. Portmap

Hay algunos servicios que funcionan sobre un mecanismo llamado Remote Procedure Call (RPC). Algunos de estos son NFS. NIS/YP, y otros. La mayor parte de los usuarios hoy en día no requieren a ninguno de ellos, y creen que basta con detener al demonio indicado. Portmap es. sin embargo, un demonio que guarda información acerca de todos estos demonios; mucha gente ignora la existencia de Portmap, sin embargo, y lo deja corriendo.

Una gran cantidad de exploits han aparecido aprovechando debilidades en el diseño de Portmap. Te sugiero fuertemente que lo desactives y elimines del sistema a menos que realmente lo requieras. Muchos sistemas lo levantan desde rcS.d, otros varios desde cada runlevel en específico.

Portmap es iniciado como demonio independiente, y escucha por el puerto 111.

r-commands

Estos servicios aparecieron cuando Internet era todavía una red académica y confiable; están hechos para permitir hacer ciertas operaciones fácilmente entre computadoras. Los principales r-commands son:

rsh. rexec. rlogin Diferentes maneras de ejecutar un comando arbitrario en una máquina remota, o iniciar una sesión en ella rcp Copia archivos de una computadora a otra rwho Revisa qué usuarios están conectados a un servidor remoto

Estos comandos son triplemente peligrosos:

Toda la interacción es transmitida en claro y puede ser facilmente detectada por un sniffer, y modificada con herramientas que cualquier cracker interesado puede fácilmente conseguir. Para la autentificación. la contraseña es transmitida en claro, como en el caso de Telnet. Tienen un mecanismo de confianza, manejado a través del archivo /etc/hosts.equiv y los archivos .rhosts en el home de cada usuario, en el que especifican en qué servidores confían. Esta confianza se basa únicamente en su dirección IP, por lo que si alguien envía paquetes que parezcan venir de dichas máquinas, éste pasará sin requerir autentificación.

Los r-commands son típicamente activados desde inetd. por lo que basta comentar las líneas que los invocan (y claro. reiniciar inetd) para cerrarlos. Típicamente vienen declarados en inetd como shell (rsh), login (rlogin), exec (rexec) y rwho (rwho).

Todos estos comandos pueden, además, ser reemplazados con sus equivalentes seguros con Secure Shell, con la misma sintaxis y prácticamente el mismo nombre

netstat

La principal preocupación son típicamente los ataques provenientes de la red1. netstat es un comando que podemos encontrar en todo Unix, y que nos permitirá cerciorarnos de que no estamos dando más servicios de los estrictamente requeridos. Esta, claro, es sólo una de las opciones que tiene netstat.

En este caso, nos interesa saber qué puertos tiene abiertos nuestro sistema. Vamos primero sobre los puertos TCP:

```
mipc:\# netstat -nap |grep -w 'LISTEN |udp'
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      828/inet
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN      227/X
tcp        0      0 0.0.0.0:1024            0.0.0.0:*               LISTEN      219/wdm
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      209/apac
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      194/sshd
tcp        0      0 0.0.0.0:515             0.0.0.0:*               LISTEN      169/lpd
udp        0      0 0.0.0.0:177             0.0.0.0:*                           219/wdm
udp        0      0 0.0.0.0:67              0.0.0.0:*                           150/dhcp
```

Manejo de los archivos de contraseñas

Hoy en día, casi todas las distribuciones presentan la posibilidad de utilizar shadow passwords y MD5. Tradicionalmente, en los sistemas Unix manejamos todas las cuentas del sistema en el archivo /etc/passwd, donde tenemos los datos generales del usuario así como su contraseña cifrada con el algoritmo DES del NIST. Este es un algoritmo de cifrado de 64 bits, excelente en su momento, pero casi trivial de tronar con el poder de cómputo actual

Debian y derivados - apt-get

Debian nos presenta una gran ventaja respecto a cualquier otra distribución: El sistema apt-get. Conviene tener la siguiente línea en el archivo /etc/apt/sources.list, que indica a apt-get dónde buscar los programas a instalar:

deb http://security.debian.org stable/updates main contrib non-free

Actualizar el sistema completo puede ser hecho simplemente con los comandos:

apt-get update: apt-get dist-upgrade

apt-get[1] tiene una gran cantidad de opciones; si bien correr esto a diario te ayudará a mantener el sisetma al día.

4.11. Herramientas de seguridad: Paquetes adicionales y proyectos interesantes

Prácticamente todas las distribuciones incluyen ya algunas de las herramientas de seguridad que en su momento parecieron tremendamente innovadoras y hoy en día son ya dadas por hecho. como es el caso de TCP-Wrappers de Wietse Venema. sin embargo, hay varias herramientas que pueden serte muy útiles. Algunas de ellas son:

4.11.1. Psionic: Portsentry y Logcheck

Ambas son parte del proyecto Abacus de Psionic.

Portsentry es un detector de barridos de puertos e intentos de conexión a puertos cerrados, que no sólo previene sino que toma acción correctiva bloqueando toda comunicación entre el posible atacante y nuestro sistema. Prácticamente todos los ataques comienzan con una fase de recopilación de información, en que el atacante intenta encontrar todos los datos posibles acerca de nuestro sistema.

Revisar las bitácoras es una de las obligaciones más importantes, pero más tediosas de un administrador de sistemas. Esto es en buena parte porque tenemos que acordarnos de hacerlo, tenemos que recordar en qué línea nos quedamos, no hay manera automática de prioritizar los mensajes, y nos toca analizar bloques bastante grandes.

[1] apt-get es un sistema tan poderoso que nos permite actualizar el sistema completo a una versión más nueva de la distribución con tan sólo indicarlo, sin romper dependencias ni crear conflictos. apt-get es gran parte de la razón por la que los miles de usuarios de esta distribución la aprecian tanto por sobre de RedHat y otras distribuciones.

Logcheck te simplifica esta tarea, enviándote a tu buzón con el intervalo que le especifiques, acomodado por prioridades, lo que llegue a tu bitácora.

Argus

Red que Supervisa las Herramientas

Argus es una red que supervisa la herramienta que utiliza un modelo cliente-servidor para capturar los datos y asociarlos en "transacciones."La herramienta proporciona la revisión del nivel de red; puede verificar la complacencia a un archivo de configuración de ruta, y la información puede ser fácilmente adaptada al análisis del protocolo, detecciones de intrusión, y a otras necesidades de seguridad. Argus está disponible en muchos sitios, incluyendo

ftp://ftp.andrew.cmu.edu/pub/argus /

swatch

Swatch, Simple WATCHer Program, es un archivo de registro filtro/monitor fácilmente configurable. Swatch supervisa archivos de registro y actúa para filtrar hacia afuera datos no deseados y tomar uno o más usuarios especificando acciones basadas en modelos del registro. Swatch está disponible de

ftp://ftp.stanford.edu/general/security-tools/swatch /

Crack

Herramientas de Autentificación/Password

Crack es un programa libre, disponible diseño para identificación, por el estándar que conjeturan las técnicas, UNIX DES encripta passwords que se pueden encontrar en diccionarios extensamente disponibles. Las técnicas especuladas estan descritas en la documentación del Crack.

Muchos administradores del sistema ejecutan el Crack como un sistema regular de procedimiento de administración y notifica a dueños de cuentas a quienes les han çrackeado"passwords.

El Crack está disponible de

ftp://coast.cs.purdue.edu/pub/tools/unix/crack /

Passwords shadow Si su sistema UNIX tiene una capacidad de password sombra, debería usarla. Bajo un sistema de password sombra, el archivo /etc/passwd no tiene passwords encriptados en el campo pasword. En cambio, los passwords encriptados se sostienen en un archivo sombra que no es mundialmente leíble. Consulte sus manuales del sistema para determinar si una capacidad de password sombra está disponible en su sistema y para obtener los detalles de cómo levantarlo y manejarlo.

Herramientas de Filtrado de servicios

Programa de capa TCP/IP El programa de capa TCP/IP proporciona la información de registro de una red adicional y le da la habilidad a un administrador del sistema de negar o de permitir el acceso de ciertos sistemas o dominios al host en el que el programa esta instalado. La instalación de este software no requiere ninguna modificación en el software existente de la red. Este programa está disponible de

ftp://ftp.porcupine.org/pub/security

Herramientas para Examinar Hosts para Vulnerabilidades Conocidas

ISS (Internet Security Scanner) ISS es un programa que interrogará a todas las computadoras dentro de un rango especíco de direcciones IP, determinando la postura de seguridad de cada una con respecto a varias vulnerabilidades comúnes del sistema. ISS está disponible de muchos sitios, incluyendo

ftp://coast.cs.purdue.edu/pub/tools/unix/iss

Para información extensa sobre ISS, vea

http://www.cert.org/advisories/CA-93.14.Internet.Security.Scanner.html

SATAN (Security Administrator Tool for Analyzing Networks)

SATAN es una herramienta de prueba y reporte que colecciona una gran variedad de información sobre los hosts conectados a una red de computadoras. SATAN está disponible de muchos sitios, incluyendo

ftp://ftp.porcupine.org/pub/security

Para información extensa sobre SATAN, vea

http://www.cert.org/advisories/CA-95.06.satan.html http://www.cert.org/advisories/CA-95.07a.REVISE

Herramientas Multi-Propósitos

COPS(Computer Oracle and Password System) COPS son una colección de programas públicamente disponibles que procuran identificar problemas de seguridad en un sistema de UNIX. COPS no intentan corregir cualquier diferencia encontrada; él simplemente produce un informe de sus resultados. Los COPS están disponibles de

ftp://coast.cs.purdue.edu/pub/tools/unix/cops

Herramientas de Control de Integridad

MD5 MD5 es un programa de checksum criptográfico. MD5 toma como entrada un mensaje de longitud arbitraria y produce como salida una "huella digital"de 128 bits o un o"mensaje asimilado"de la entrada. Se piensa para ser computacionalmente no factible para producir dos mensajes teniendo el mismo mensaje asimilado o para producir cualquier mensaje que tiene un objetivo específico dado en el mensaje asimilado. MD5 se encuentra en RFC 1321. Vea

ftp://coast.cs.purdue.edu/pub/tools/unix/md5

ifstatus

El programa ifstatus puede correrse en los sistemas UNIX para identificar interfaces de red que estén en depuración o en modo promiscuo. Las interfaces de red en estos modos pueden ser una señal que un intruso está supervisando la red para robar passwords y otras estaciones (vea CERT Advisory CA-94.01).

El programa no imprime ninguna salida (a menos que -v este dada) a menos que encuentra las interfaces en "malos"modos. Así que, es fácil correr el ifstatus del cron una vez por hora mas o menos. Si tiene un cron moderno que manda por correo el rendimiento de trabajos del cron a su propietario, utilice una linea como esta:

```
00 * * * * / el usr / local / el etc / el ifstatus
```

Si tiene una versión de cron que no hace esto, utilice el shell script run-ifstatus"(revise el script para utilizar la ruta correcta para el comando):

```
00 * * * * / el usr / local / el etc / corra - el ifstatus
```

ifstatus está disponible en muchos sitios, incluyendo

ftp://coast.cs.purdue.edu/pub/tools/unix/ifstatus /

smrsh

Con todas las versiones de sendmail, nosotros recomendamos que usted utilice el programa shell restringido de sendmail, smrsh, creado por Eric Allman (el autor original de sendmail). Cuando está configurado correctamente, el programa smrsh puede ayudar a proteger contra una vulnerabilidad que pueda permitir que los usuarios remotos o locales desautorizados ejecuten programas como cualquier usuario del sistema con excepción de raíz. Por ejemplo, el smrsh puede evitar que un intruso use los tubos (—) para ejecutar comandos arbitrarios en su sistema.

Nosotros animamos a que utilice el smrsh sin importar si usted utiliza el sendmail provisto del vendedor o instala el sendmail usted mismo, y sin importar las correcciones que han estado instaladas.

Comenzando con la versión 8,7,1 del sendmail, el smrsh se incluye en la distribución del sendmail, en el subdirectorio smrsh. Vea el archivo RELEASE_NOTES para una descripción de cómo integrar el smrsh en su archivo de configuración del sendmail.

El smrsh también está disponible de muchos sitios, incluyendo

http://www.sendmail.org/

Advertencia: Si usted está ejecutando una versión vieja del sendmail tal vez usted deba instalar el smrsh por separado, los intrusos continuarán pudiendo explotar las vulnerabilidades que fueron fijadas en versiones posteriores de sendmail. Le instamos a que actualize la versión del correo del sendmail y después ejecute las herramientas, que se incluyen con la distribución. Refiérase a los archivos siguientes para información extensa sobre el smrsh y sendmail:

http://www.cert.org/advisories/CA-96.20.sendmail_vul.html http://www.cert.org/advisories/CA-96.24.sendmail.daemon.mode.html http://www.cert.org/advisories/CA-96.25.sendmail_groups.html mail.local Algunas versiones de /bin/mail basadas en BSD 4.3 UNIX son vulnerables debido a la sincronizació de Windows en la forma en que /bin/mail utiliza los directorios escribibles. Si usted no puede instalar un parche de su vendedor, reemplace /bin/mail con mail.local. Empezando con el sendmail versión 8.7.1, mail.local esta incluido en la distribución del sendmail, en el subdirectorio mail.local. El programa también está disponible de

http://www.sendmail.org/

Para información completa aceca de mail.local, vea

http://www.cert.org/advisories/CA-95.02.binmail.vulnerabilities.html

Otra Lectura Sobre las Herramientas de Seguridad Para una lista adicional de herramientas de seguridad, vea el Apéndice B del "Lista de Control de Seguridad de Computadoras UNIX"desarrolladas por la Australian Computer Emergency Response Team (AUSCERT). Una copia de la lista de control de AUSCERT puede encontrarse en

ftp://ftp.auscert.org.au/pub/auscert/papers/unix_security_checklist

4.11.2. Comunicación

Las redes hoy en día son inseguras, y nos imposibilitan confiar en lo que transportan. El cifrado a todos niveles nos permite asegurar la Privacidad Integridad Irrefutabilidad . Las herramientas principales para cifrar nuestra comunicación son:

Secure Shell

Una gran ventaja de los sistemas Unix es su capacidad de ser administrados remotamente. Por muchos años,

Secure Shell nos permite hacer todo esto, de una manera segura y cifrada. Pensado incluye a scp y sftp, para la copia de archivos remotos, y dentro del mismo ssh, el mecanismo para crear túneles de puertos sobre canales cifrados, lo que nos permite, entre otras muchas cosas, utilizar sesiones X remotamente sin preocuparnos por intrusos ssh nos puede dar toda la infraestructura necesaria para una VPN completa.

stunnel

Algunas aplicaciones hoy en dia, no pueden ser manejadas por Secure Shell. Un ejemplo muy clásico es el de las máquinas Windows cliente que consultan su correo en nuestro servidor podemos poner a su disposición versiones cifradas con el estándar SSL de los protocolos inseguros que manejan. La mayor parte de los clientes de correo reconocen los puertos 993 y 995 para IMAP y POP3 (respectivamente) sobre SSL. Stunnel nos permite cifrar cualquier protocolo, siempre que el programa cliente comprenda también SSL.

PGP

Probablemente el programa de cifrado más conocido en el mundo sea PGP. Desde su polémica aparición hace 10 años ha permitido a todo mundo tener acceso a criptografía fuerte junto con un robusto esquema de redes de confianza para llaves públicas, nos permite mantener nuestras comunicaciones cifradas y seguras.

4.12. Herramientas para el administrador

Simplifican nuestra tarea de administrar un equipo en diferentes rubros. ¿Ejemplos? logcheck, npasswd, passwd+, crack, etc. Para robustecer el sistema existen herramienta como:

OpenWall

El proyecto OpenWall es un grupo de parches disponibles para el kernel del Linux, y una muy buena forma de prevenir ataques como Buffer Overflows y similares. Son una coleccion de mejoras de seguridad para integrar en bloque en el kernel, configurables todas ellas desde una nueva sección de Seguridad que es añadida al menu de configuración del kernel, y que verás a la hora de reconfigurarlo. Estos parches estan disponibles para diferentes versiones del kernel, pero hay que recalcar que no son parte de la distribución central. algunos de sus puntos son:

Stack no ejecutable: La mayor parte de los ataques por buffer overflow buscan sobrescribir la direccion de retorno en la pila, apuntando a codigo arbitrario introducido por el atacante, que tambien es puesto en la pila. Si el area de la pila no es ejecutable, estos buffer overflows se vuelven mas complicados de explotar. Ligas y FIFOs en /tmp restringidas: Ciertos programas son vulnerables a que el intruso cree una liga simbólica en /tmp, la cual el programa no comprueba, y el atacante usa para sobreescribir u obetner datos de otra región del sistema. Activando esta opcion se reduce el impacto de este tipo de ataques, no permitiendo a un proceso seguir un archivo que es un enlace en un directorio temporal

/proc restringido: restringe el acceso a los directorios en /proc, de tal forma que los usuarios solo pueden ver sus procesos en el sistema y ningun tipo de actividad de conexiones de la red, salvo que se encuentren en un grupo especifico. Tambien impide el uso del comando 'dmesg' a los usuarios. Destruir segmentos de memoria no utilizados: Unix te permite especificar cuanta memoria puede consumir un proceso. Desafortunadamente los segmentos de memoria compartidos pueden existir sin estar asociados a ningun tipo de proceso. Esto dèstruira segmentos de memoria que ya no se encuentran en uso, que no han sido vinculados a ningun proceso.

4.12.1. Livecd

Son CD-Rom que contienen sistema operativos y su ventaja es que no modifican al sistema, a continuación daré algunos ejemplos:

Knoppix STD 0.1b

STD (Security Tools Distribution) es una versión personalizada de Knoppix, una distribución de Linux pensada para ser ejecutada directamente desde el CD-ROM ("Live CD"). Utiliza el núcleo 2.4.20 y KDE 3.1, da soporte a una gran cantidad de dispositivos de hardware (que son detectados y configurados automáticamente). Cuando se arranca la máquina con Knoppix STD, no se realiza ningún tipo de modificación en la configuración del computadora.`http://www.knoppix-std.org/`

Knoppix STD incluye un gran número de herramientas de interés para la seguridad, todas ellas preparadas para ser ejecutadas directamente desde el CD. Las herramientas se dividen en varias categorías: autenticación, identificación de contraseñas, cifrado; herramientas para el análisis forense, cortafuegos, honeypots, sistemas de detección de intrusiones; herramientas para la gestión de redes; un gran número de herramientas para la realización de pruebas de penetración, sniffers; herramientas para la realización de valoraciones de seguridad y herramientas para la realización de pruebas de redes sin fines.

LocalAreaSecurity 0.4

Esta es otra distribución "LiveCD", de pequeño tamaño (185 MB, pensada para instalarse en un CD pequeño, de la medida de una tarjeta de crédito). También está basada en Knoppix y utiliza el núcleo 2.4.20.`http://www.localareasecurity.com/`

LocalAreaSecurity está especializada en la realización de pruebas de verificación de la seguridad y en las pruebas de penetración, incluyendo un gran número de herramientas especializadas: sniffers, cifrado, monitorización de redes, detección de información oculta, obtención de información, etc.

Phlax (Profesional Hacker's Linux Assault Kit) 0.1

Otra distribución "Live CD"que se ejecuta directamente desde el CD-ROM. Está especializada en la realización de análisis de seguridad, pruebas de penetración, análisis forense y auditorios de seguridad. Entre las herramientas incluidas encontramos: sniffers y herramientas para el análisis del tráfico capturado, herramientas para el análisis de protocolos y del funcionamiento del sistema, extracción de datos de sistemas de archivos, cifrado de archivos, etc.`http://www.phlak.org/`

R.I.P. (Recovery Is Posible) Linux

Se trata de una distribución de Linux pensada por recuperar datos de sistemas de archivos defectuosos. Merced a esta distribución, el autor de este boletín pudo recuperar los datos de una máquina con un disco duro defectuoso que Windows XP se negaba a reconocer ni tampoco sabía como reparar. Con R.I.P. Linux fue posible no tan solo montar y acceder a la información, sino transferirla por la red a otro sistema.`http://www.tux.org/pub/people/kent-robotti/looplin`

R.I.P. Linux funciona directamente desde el CD-ROM y da soporte a diversos sistemas de archivos: ext2, ext3, reiser, jfs, xfs, ufs, NTFS, FAT16 y FAT32

WARLINUX 0.5

http://sourceforge.limpio/projects/warlinux/ Esta distribución de Linux, en modo texto, está especialmente pensada para la verificación de la seguridad de las redes inalámbricas. Funciona directamente desde el CD y permito identificar las redes inalámbricas que están al alcance del computadora y la realización de auditorios de seguridad y valoración de su nivel de seguridad.

FIRE

Está versión de Linux incluye las herramientas necesarias para la realización de valoraciones de seguridad, respuesta a incidentes de seguridad, pruebas de penetración y análisis forense de sistemas y recuperación de datos en sistemas Windows, Solaris (SPARC) y Linux (x86). Adicionalmente, FERIO incluye un programa para la detección de virus (F-Prot).`http://biatchux.dmzs.comc`

Otras distribuciones similares a estas que hemos comentado son Penguin Sleuth Kit, @stake Pocket Security Toolkit v3.0, ThePacketMaster Linux Security Server y Trinux.

Capítulo 5

Conclusiones

Al concluir este trabajo .de tesis, corroboro la necesidad de implementar seguridad en cómputo, indistintamente, si se quiere otorgar servicios a una red local o a Internet como caso particular. Y dando como punto principal

La implementación y uso de políticas de red , que se explicó en el capítulo 1 y se retomo en el cuatro. Dada su necesidad y mantenimiento al documento y a la usabilidad técnica y practica.

La seguridad física es otro aspecto que se manejo modestamente en el capitulo 4. El uso combinado de las bitácoras que registran el momento en que un posible intruso está delante de una computadora y la evidencia fílmica, solo en caso de que existan cámaras.

Personal confiable Un problema fuertísimo. El cual no se toca en el trabajo, pero deberá ser perseguido por expertos en la psicología. Solo se pudo comentar de que personas nos tendríamos que proteger en el capitulo 2.

La Administración del sitio o servidor de información Recomiendo ampliamente la programación para labores triviales. Es la automatización de la administración. Si hay alguna actividad que hay que realizar todos los días o periódicamente, la mayor parte de los administradores la olvidaría cotidianamente por estar en otras de las tantas actividades que nuestro trabajo nos requiere. El respaldo diario y la ejecución periódica de ciertos programas son perfectos candidatos para la automatización. Dependiendo qué tanto queramos hacer esto implicará más o menos programación.

Automatización se podría al menos generar respaldos. una parte importante de la generación de respaldos es la revisión de los mismos, para no llevarse sorpresas de la in

funcionalidad de los mismos en un caso crítico

Actualización por primera instancia. Es un hecho que aunque a los sistemas operativos y aplicaciones de una red se le apliquen los más recientes parches de seguridad éstos siguen expuestos al ataque de hackers avanzados, ya que al momento en que un parche es recién liberado es porque se tenían ya semanas de que un hacker había violado la seguridad precisamente a través del hoyo que el parche pretende tapar.

Aprendizaje continuo Cada día aparecen nuevos inventos, desarrollos, programas y riesgos en el campo de la computación, y el administrador responsable debe estar al tanto de tantos como le sea posible. Importancia de los servicios de noticias en tecnología, listas de correo y revistas. Para estar al tanto de los acontecimientos conforme se van dando, no hay como leer estas fuentes de información. Así mismo, la dificultad de encontrar gente con experiencia en el análisis de ataques y la falta de un marco de trabajo común, favorece que muchos de estos delitos no sean denunciados y, por supuesto, no sean penados.

Análisis Forense La parte fuente del trabajo. Al hacer uso de técnicas y herramientas que nos ayuda a visualizar e problema de haber sido vulnerados y que esta en riesgo el sistema o la confiabilidad de los datos y servicios que deberán de estar bajo nuestro control (el administrador).

Para alargar el tiempo de ser vulnerados se tendría que monitorear, ya sea equipos trampa esto si se desea conocer al enemigo que nos acosa, esto con la virtud de que no sean factor de riesgo para nuestro sitio.

Auditorias de seguridad Las empresas o instituciones conocen, o han de conocer todos estos problemas. Y cuando deciden darles una solución global, buscando los puntos débiles de su seguridad para atajarlos de una vez, pueden decidirse por una Auditoria de Seguridad. Las auditorias son actividades muy comunes en estos entornos empresariales, especialmente las realizadas por personal externo, y permiten conocer el nivel de seguridad y las acciones a emprender para corregir los posibles fallos.

El hecho de que en general las auditorias las realicen personas externas, permite mantener un nivel de objetividad que muchas veces no se da entre el personal propio, por razones obvias.

Una auditoria puede durar, en función del tamaño del sistema, desde unos pocos días, hasta varias semanas. En general siguen normas estrictas y protocolos extensos y requieren fuertes compromisos de apoyo de los recursos internos de la organización en cuestión. Pueden, sin embargo, ser más flexibles. Es evidente que el peligro no acecha sólo a las grandes corporaciones y que no sólo este tipo de negocios son los negocios importantes, menos aún para los propietarios de los negocios amenazados, por pequeños que sean.

El costo de una buena gestión de seguridad siempre es menor que el valor que pueden tener los datos internos de la empresa. La auditoria de seguridad es uno de los servicios llamados a un mayor desarrollo en los próximos años. El desarrollo de Internet es espectacular y las posibilidades del comercio electrónico son ilimitadas.

El proceso comienza con un análisis de las amenazas potenciales que enfrentan a una organización. Examina sistemas, políticas y practicas de la organización para identificar sus vulnerabilidades. El análisis continúa con una valoración de riesgo y concluye con un informe de valoración y una serie de recomendaciones.

Con respecto a los costos de la seguridad se puede pensar que son elevados, y en muchos casos los son, especialmente cuando se trata de una auditoria convencional, aunque evidentemente existen diferencias entre las necesidades de cada caso, que se relacionan de forma directa con el costo.

Las políticas de seguridad tal y como la palabra lo dice, se asemejan a los seguros de la vida cotidiana, muchas veces no se toma una decisión al respecto hasta que no se conoce un caso cercano a quien la adversidad le coge por sorpresa. La seguridad representa un gasto que muchas veces parece inútil y que se podría evitar, aunque el costo de una buena gestión de seguridad siempre es menor que el valor que pueden tener los datos internos de la empresa.

De una forma o de otra, es evidente que el comercio electrónico es el futuro para gran parte de la actividad económica, y que éste es imposible si no se resuelven los problemas de seguridad en la red. Para eso están empresas como IPS Seguridad, para garantizar que la seguridad en Internet sea posible.

El campo de trabajo es intangible. Esto hace necesario desarrollar técnicas y adaptar los existentes métodos de forma tal de circunscribir nuestro trabajo de conseguir información conocimiento dentro de un marco de seguridad. Redundando al capitulo 4 de Reforma, describo profundamente los siguientes tópicos:

DISEÑO SEGURO REQUERIDO Cuando se diseña un sistema se lo hace pensando en su Operatividad Funcionalidad dejando de lado la Seguridad Será necesario establecer una correspondencia y pertenencia entre las técnicas adoptadas conformando un sistema de seguridad: y no procedimientos aislados que contribuyan al caos general existente. Esto sólo puede lograrse al integrar la seguridad desde el comienzo, desde el diseño, desde el desarrollo.

LEGISLACIÓN VIGENTE Las tecnologías involucradas en estos procesos condicionan las técnicas empleadas, los tiempos condicionan esas tecnologías y, paradójicamente, las legislaciones deben adaptarse a los rápidos cambios producidos. Esto hace obligatorio

no legislar sobre tecnologías actuales, sino sobre conceptos y abstracciones que podrán ser implementados con distintas tecnologías en el presente y el futuro. Es urgente legislar un marco legal adecuado, no solo que castigue a los culpables sino que desaliente acciones hostiles futuras.

TECNOLOGÍA EXISTENTE Existen infinidad de métodos (muchas veces plasmados en herramientas) que permiten violar un sistema. El profesional cuenta con la misma tecnología para la evaluación de la seguridad del bien a proteger y otras pensadas para la protección como fin. Esto hace que muchas veces, la seguridad, sea asunto de la idoneidad del profesional. En algunos campos, la Tecnología deberá ampararnos ante la desaparición de elementos naturales. Por mencionar un ejemplo: la firma digital (Tecnología Criptográfica) debe cubrir la brecha que deja la inexistencia de la firma caligráfica en archivos de información.

DAÑOS MINIMIZABLES Algunos pocos métodos realmente novedosos de infiltración ponen en jaque los sistemas de seguridad. Aquí, se prueba la incapacidad de lograr 100 % de seguridad, pero también es hora de probar que los riesgos, la amenaza, y por ende los daños pueden ser llevados a su mínima expresión. Muchas veces vasta con restringir accesos a información no utilizada o que no corresponde a los fines planteados. Otras veces la capacitación será la mejor herramienta para disminuir drásticamente los daños.

RIESGOS MANEJABLES La Seguridad Perfecta no existe, y de hecho dudo que algún día exista, pero los riesgos deben y pueden ser manejables.

COSTOS El costo en el que se incurre suele ser una frusleria comparados con aquellos luego de producido un daño. El desconocimiento y la falta de información son el principal inconveniente cuando se evalúa la inclusión de seguridad como parte de un sistema.

PERSONAS INVOLUCRADAS El desarrollo de software es una "ciencia" imperfecta; y como tal es vulnerable. Es una realidad, y espero haberlo demostrado en el extenso capítulo de "Amenazas Humanas", que la Seguridad involucra manipulación de naturaleza humana. Es importante comprender que:

1. La Seguridad consiste en Tecnología y Política. Es decir que la combinación de la Tecnología y su forma de utilización determina cuan seguros son los sistemas.
2. El problema de la Seguridad no puede ser resuelto por única vez. Es decir que constituye un viaje permanente y no un destino.
3. En última instancia la Seguridad es una serie de movimientos entre "buenos" y "malos".

El uso de herramientas comerciales y gratuitas nos podrán permitir establecer un esquema integral de seguridad después de un análisis sobre las necesidades y recursos para su funcionalidad.

Por ejemplo, una herramienta básica, los firewall que permiten aislar la red interna de la externa, con control del tipo de protocolo que circula y su origen y destino. Y para fortalecimiento de la seguridad, la criptografí, por ejemplo, los sistemas de correo basados en cualquiera de los programas que utilizamos habitualmente pueden complementarse con mecanismos de encriptación de datos y firma electrónica, ya sea utilizando protocolo S/MIME o PGP.

Dado este panorama. Hoy no se puede decir que la conexión a Internet o a cualquier otra red abierta no se pueda realizar de forma segura, existen las herramientas y la mayoría de ellas seguro que se encuentran incorporadas en el sistema operativo de sus servidores y estaciones de trabajo.

Las consecuencias de un mal diseñó de red y de seguridad, de la no utilización de herramientas adecuadas y el desconocimiento de lo que le puede estar pasando a nuestra red, son los peores enemigos de cualquier sistema.

Bibliografía

[1] *http://200.44.120.106/Volumes/internetbib/curso/cap1/cap1tem1.htm.*

[2] *http://cert.org.*

[3] *http://es.conectiva.com.*

[4] *http://odci.gov/cia.*

[5] *http://project.honeynet.org.*

[6] *http://snorticus.baysoft.net/.*

[7] *http://trinux.sourceforge.net/.*

[8] *http://www. gocsi.com.*

[9] *http://www.amazon.com.*

[10] *http://www.andrew.cmu.edu/rdanyliw/snort/snortacid.html.*

[11] *http://www.bastille-linux.org/.*

[12] *http://www.caldera.com.*

[13] *http://www.debian.org.*

[14] *http://www.demolinux.org/.*

[15] *http://www.eeye.com.*

[16] *http://www.freecolormanagement.com/lods/.*

[17] *http://www.fsf.org.*

[18] *http://www.gwolf.cx/seguridad/logcheck/.*

[19] *http://www.gwolf.cx/seguridad/portsentry/.*

[20] *http://www.ibm.com.*

[21] *http://www.icsa.net.*

[22] *http://www.itm.tu-clausthal.de/perle/hal91/.*

[23] *http://www.kriptopolis.com.*

[24] *http://www.lafirmadigital.com.*

[25] *http://www.linuxrouter.org.*

[26] *http://www.mxcert.org.mx.*

[27] *http://www.mxcert.org.mx/recursos.*

[28] *http://www.nai.com.*

[29] *http://www.nasa.gov.*

[30] *http://www.nextvision.com.*

[31] *http://www.nist.gov.*

[32] *http://www.nl.gob.mx/pagina/Enlaces/ciapem/Des_sis.htl.*

[33] *http://www.ods.com.ua/win/eng/security/Max_Security http://www.itlibrary.com/reference/library/*

[34] *http://www.openbsd.org.*

[35] *http://www.openbsd.org.mx.*

[36] *http://www.openwall.com/.*

[37] *http://www.pgpi.com-http://www.pgpinternational.com-http://pgp.org.*

[38] *http://www.psionic.com/abacus.*

[39] *http://www.redhat.com.*

[40] *http://www.rediris.es-http://www.rediris.es/cert.*

[41] *http://www.rsa.com.*

[42] *http://www.rt.com/man/ipchains.8.html.*

[43] *http://www.securityfocus.com.*

[44] *http://www.seguridadcorporativa.org.*

[45] *http://www.seguridad.unam.mx/Tutoriales/tutoriales.html.*

[46] *http://www.seguridata.com.*

[47] *http://www.slackware.com.*

[48] *http://www.snort.org/.*

[49] *http://www.sophos.com.*

[50] *http://www.spirit.com/csi.*

[51] *http://www.sun.com.*

[52] *http://www.suse.com.*

[53] *http://www.telematik.informatik.uni-karlsruhe.de/lehre/seminare/LinuxSem/downloads/netfilter/ipt HOWTO.html.*

[54] *http://www.toms.net/rb/.*

[55] *http://www.tripwire.org/.*

[56] *http://www.truesecure.com.*

[57] *http://www.vanhackez.co/set - http://www.thepentagon.com/paseante.*

[58] *http://www.verisign.com.*

[59] *http://www.w3.com.*

[60] *http://www.zelow.no/floppyfw/.*

[61] *http://www.zonelabs.com.*

[62] *redhat-announce-list-admin@redhat.com.*

[63] *RFC 1180: TCP/IP Tutorial. T. Socolofsky - C. Kale. Enero 1981.*

[64] *RFC 1244: Site Security Handbook. J. Reynolds - P. Holbrook. Julio 1991.*

[65] *RFC 1812: Requirements for IP Version 4 Routers. F. Baker. Junio 1995.*

[66] *RFC 1939: Post Office Protocol-Version 3. J. Myers - M. Rose. Mayo 1996.*

[67] *RFC 2045-2049: Multipurpose Internet Mail Extensions: MIME. N. Freed - N. Borenstein. Noviembre 1996.*

[68] *RFC 2196: Site Security Handbook (reemplaza a RFC 1244). B. Fraser. Septiembre 1997.*

[69] *RFC 2401: Security Architecture for the Internet Protocol. S. Kent - R. Atkinson. Noviembre 1998.*

[70] *RFC 2411: IP Security Document Roadmap. R. Thayer - N. Doraswamy - R. Glenn. Noviembre 1998. Informational,.*

[71] *RFC 2440: OpenPGP Message Format. J. Callas - L. Donnerhacke - H. Finney - R. Thayer. Noviembre 1998.*

[72] *RFC 2459: RFC Internet X.509 Public Key Infrastructure-Certificate and CRL Profile. R. Housley - W. Ford - W. Polk - D. Solo. Enero 1999.*

[73] *RFC 2527: Internet X.509 Public Key Infrastructure-Certificate Policy and Certification Practices Framework. S. Chokhani-W. Ford. Marzo 1999.*

[74] *RFC 2577: FTP Security Considerations. Network Working Group M: Allman. Mayo 1999.*

[75] *RFC 2588: IP Multicast and Firewalls. R. Finlayson. Marzo 1999.*

[76] *RFC 2828: Internet Security Glossary. R. Shirey. Mayo 2000.*

[77] *Departmen of defense trusted computer. System evaluation criteria (Orange Book).* Departmant of Defense Standard., 1985.

[78] *Network security and access control. Network Access Special Interest Group.* ISSA, 1988.

[79] *http://www.unixsup.com/unixlinux/historiaunixcuxs.html,* 2002.

[80] Gustavo. Miguel ALDEGANI. *Seguridad Informática.* MP Ediciones. Uruguay. 1997.

[81] ANONIMO. *Máxima Seguridad en Internet.* Editorial Anaya Multimedia, 1998.

[82] N. Derek Arnold. *Unix Security: A Practical Tutorial.* McGraw Hill, 1993.

[83] Andre' Bacard. *Computer Privacy Handbook.* Peachpit Press. 1995.

[84] Rebecca BACE. *An Introduction to Intrusion Detection.* Infidel Inc.for ICSA Inc, EEUU. 1999.

[85] Richard M. Baker. *Network Security: How to Plan For It and How to Achieve It.* McGraw-Hill, Inc.

[86] P. Baran. *On Distributed Communications Networks.* IEEE, 1964.

[87] L. Todd Heberlein Biswanath Mukherjee and KarlÑ. Levit. *Network intrusion detection.* fEEE Network. Mayo/Junio 1994.

[88] SCHNEIER Bruce. *Applied Cryptography.* Editorial John Wiley & Sons., EEUU. 1995.

[89] SCHNEIER Bruce. *Secrets & Lies.* John Wiley & Sons, EEUU. 2000.

[90] Richard Bryant. *Unix Security for the Organization.* Sams, 1994.

[91] V. G. y R. E. Kahn. Cerf. *A Protocol for Packet Network Interconnection.* IEEE, 1974.

[92] D. Brent Chapman and Elizabeth D. Zwicky. *Building Internet Firewalls.* O'Reilly and Associates, Inc., 1995.

[93] William Cheswick and Steven Bellovin. *Firewalls and Internet Security.* Addison Wesley, 1994.

[94] Stephen Cobb. *The Stephen Cobb Complete Book of PC and LAN Security.* Windcrest Books, 1992.

[95] Dr. Frederick B. Cohen. *Protection and Security on the Information Superhighway.* John Wiley & Sons, 1995.

[96] Hugo Cornwall. *The Hacker's Handbook.* E. Arthur Brown Company.

[97] David A. Curry. *Improving the security of your UNIX system.* SRI, 1990.

[98] David A. Curry. *Unix System Security: A Guide for Users and Systems Administrators.* Addison-Wesley, 1992.

[99] Karl Seger David Icove and William VonStorch. *Computer Crime: a Crimefighters Handbook.* O'Reilly & Associates, 1995.

[100] David R. Spafford David K. Hess and Udo W. Pooch. *A UNIX network protocol security study: Network information service.* Texas A and M University, 1991.

[101] d.brent chapman. *construya firewalls para internet.* orailly, 1997.

[102] Thane Frivold Debra Anderson and Alfonso Valdes. *Next-generation intrusion detection system (NIDES).* Technical report, SRI International, 1995.

[103] Dorothy Denning. *Cryptography and Data Security.* Addison-Wesley Publishing Co., 1982.

[104] Daniel Farmer and Eugene H. Spafford. *The COPS security checker system.* Computer Emergenci Response Team, 1995.

[105] Dan Farner and Wietse Venema. *Improving the security of your site by breaking into it.* Sun Microsystems, 1986.

[106] Rik Farrow. *Unix System Security.* Addison Wesley, 1991.

[107] Carlos M. FERNADEZ. *Seguridad en Sistemas Informáticos.* Ediciones Díaz de Santos S.A., España. 1988.

[108] Philip Fites and Martin Kratz. *Information Systems Security.* Van Nostrad Reinhold, 1993.

[109] Karen Forcht. *Computer Security Management.* Boyd and Fraser, 1994.

[110] Jeremy Frank. *Artificial inteligence and intruson detection: Current and future directions.* University of California at Davis, 1994.

[111] Helen Fouche Gaines. *Cryptanalysis, a study of ciphers and their solution.* Dover Publications, 1956.

[112] Simson Garfinkel. *PGP: Pretty Good Privacy.* O'Reilly & Associates, Inc., 1995.

[113] Simson Garfinkel and Gene Spafford. *Practical Unix Security.* O'Reilly & Associates, Inc., 1991.

[114] Ribagorda garnach. *Seguridad en unix.* españa, 1996.

[115] Morrie Gasser. *Building a Secure Computer System.* Van Nostrand Reinhold Co., New York., 2000.

[116] Katie Hafner and John Markoff. *Cyberpunk.* Simon and Schuster, 1991.

[117] Kaihu Chen Henry S. Teng and Stephen C Lu. *Analysis using inductively generated predicted rules. Technical report.* Piscataway, 1990.

[118] Lance Hoffman. *Modern Methods for Computer Security.* Prentice Hall, 1977.

[119] IEEE, editor. *Symposium on Reserach in Computer Security and Privacy.* IEEE, 1990.

[120] Koral Ilgun. *USTAT: A real-time intrusion detection system for UNIX.* University of Mastersacute: thesis, California Sànta Barbara, 2003.

[121] R Kahn. *Communications Principles for Operating Systems, Memorandum interno BBN.* 1972.

[122] L. Kleinrock. *Information Flow in Large Communication Nets.* 1961.

[123] Sandeep Kumar and Eugene H. *An application of pattern matching in intrusion detection. Technical report. The COAST project.* Spafford University of California, 1994.

[124] KLANDER Lars. *A Prueba de Hackers.* Editorial Anaya Multimedia, EEUU. 1998.

[125] Ricardo LEVIN. *Virus Informáticos.* Mc Graw Hill., España. 1992.

[126] Steven Levy. *Hackers: Heroes of the Computer Revolution.* Doubleday, 1984.

[127] Mark Ludwig. *The Little Black Book of Computer Viruses.* American Eagle Publications, 1990.

[128] Mark A. LUDWIG. *The Little Black Book of Computer Viruses.* Electronic Edition. American Eagle Publications. Inc.

[129] John McAfee and Colin Haynes. *Threats to Your System.* St. Martin's Press, 1989.

[130] Joel. KURTZ George. MCLURE, Stuart. SCAMBRAY. *Hackers, Secretos y Soluciones para la Seguridad en Redes.* Osborne McGraw-Hill., España. 2000. http://www.hackingexposed.com.

[131] Sun Microsystems. *Administering Security, revision A.* 1990.

[132] Sun Microsystems. *System Network Administration, revision A.* 1990.

[133] Sun Microsystems. *SunOS Reference Manual, part number 800-1751-10.* Sun Microsystems, Mayo 1988.

[134] Peter G. Neumann. *Computer Related Risks.* Addison-Wesley, 1995.

[135] National Institute of Standards and Technology. U.S. *An introdution to Computer Security: The NIST Handbook.*. Department of Commerce., 1994.

[136] Jerry Papke. *Combatting Computer Crime.* McGraw-Hill, Inc. / Chantico Publishing Company, Inc.. 1992.

[137] Charles P. Pfleeger. *Security in Computing.* Prentice Hall, 1989.

[138] A. Z. Tirkel R. G. van Schyndel and C. F. Osborne. *A digital watermark. In International Conference on Image Processing, volume 2, pages 86-90.* IEEE, 1994.

[139] A. Maccabe R. Heady, G. Luger and M. Servilla. *The architecture of a network level intrusion detection system.* University of New Mexico, 1990.

[140] Markus J. Ranum. *Thinking about firewalls. Technical report, Trusted.* Information Systems Inc., 1994.

[141] Eric S. Raymond. *The New Hacker's Dictionary.* Adisson Wesley, 1995.

[142] Dennis Ritchie. *On the security of UNIX. Technical report.* AT&T Bell Laboratories, 1979.

[143] Lic. Ruben. Seguridad en unix para administradores. In *Seg. Unix.*

[144] Deborah Russell and G.T. Gengemi Sr. *Computer Security Basics.* O'Reilly & Associates, Inc., 1991.

[145] Crocker S. *RFC001 Host software.* 1969.

[146] Syed M. Sarwar. *El libro de UNIX.* 2000.

[147] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* John Wiley & Sons, 1994.

[148] Bruce Schneier. *E-Mail Security with PGP and PEM.* John Wiley & Sons, 1995.

[149] Schubar. *Weackness in the DNS, the domain name service. Technical report.* Purdue University, 1993.

[150] Karanjit S. Siyan and Chris Hare. *Internet Firewalls and Network Security.* New Riders Publishing, 1995.

[151] Stephen E. Smaha. *Haystack: An intrusion detection system. In In Fourth Aerospace Computer Security Applications Conference.* 1988.

[152] Martin Smith. *Commonsense Computer Security.* McGraw-Hill, 1993.

[153] Steven R. Snapp and Stephen E. Smaha. *Signature analysis model definition and formalism.* Agosto 1992.

[154] William Stallings. *Protect Your Privacy: A Guide for PGP Users.* Prentice-Hall, 1994.

[155] William Stallings. *SNMP, SNMPv2 and CMIP. The Practical Guide to Network-Management Standards.* Addison-Wesley Publishing Company, Abril 1994.

[156] William. STALLINGS. *Network And Internetwork Security.* Prentice Hall, EEUU. 1998.

[157] Ingo Stengel. *Security architectures based on active firewall components, 1998.* FH-Darmstadt.

[158] Bruce Sterling. *The Hacker Crackdown: Law and Disorder on the Electronic Frontier.* Bantam Books, 1982.

[159] Bruce. STERLING. *La Caza de Hackers. Freeware Literario.* http://www.kriptopolis.com, 1999.

[160] W. Richard Stevens. *TCP/IP Illustrated Volume I: The Protocols.* Addison Wesley, 1994.

[161] Cliff Stoll. *The Cuckoo's Egg.* Simon and Schuster, 1989.

[162] Cliff Stoll. *The Cuckoo's Egg.* Doubleday, 1989.

[163] Andrew S. Tanenbaum. *Unix in 1988.* Adisson Wesley, 1988.

[164] Andrew S. TANENBAUM. *Redes de Computadoras.* Prentice-Hall., EEUU. 1997.

[165] Chris Tomlinson. *Solaris security: A practical guide. Technical report.* Sun Microsystems, 1994.

[166] Wietse Venema. *TCP Wrapper: Network monitoring, access control and booby traps. In Proceedings of the 3rd USENIX Unix Security Symposium, pages 85-92.* The USENIX Association, Septiembre 1992.

[167] M. J. Williamson. *Thoughts on cheaper Non-Secret encryption. Technical report.* CESG, Agosto 1976.

[168] M. J. Williamson. *Non-Secret encryption using a finite field. Technical report.* CESG, Enero 1974.

[169] Patrick H. Wood and Stephen G. Kochan. *Unix System Security.* Hayden Books, 1985.

[170] Dave Wreski. *Linux Security Administrator?s Guide.* http://nic.com/ dave/Security/, 1998.

[171] Tatu Ylonen. *SSH - Secure login connetions over the Internet. In Proceedings of the 6th USENIX Security Symposium, pages 37-42.* The USENIX Association, Julio 1996.

[172] Phil Zimmermann. *The Official PGP User's Guide.* M.I.T. Press, 1995.

[173] Phil Zimmermann. *PGP: Source Code and Internals.* M.I.T. Press, 1995.

[174] Jeffrey B. Zurschmeide. *IRIX Advanced Site and Server Administration Guide. Technical Report 007-0603-100.* Silicon Graphics, Inc, 1994.