UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

"ARAGÓN"

DETECCIÓN DE INTRUSOS EN LINUX

T E S I S

PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

HÉCTOR MANUEL RODRÍGUEZ RANGEL

DIRECTOR DE LA TESIS: M. EN C. MARCELO PÉREZ MEDEL

ESTADO DE MÉXICO **2003**

A

PAGINACION

DISCONTINUA

TESIS
CON
FALLA DE
ORIGEN

A la Universidad, por brindarme la formación académica necesaria para desarrollar esta tesis y continuar mis estudios. Además de agradecer a mis revisores de tesis por ayudarme a ser mejor profesionalmente.

A mis padres, por darme su apoyo para realizar las metas que me he propuesto.

Al Centro Tecnológico Aragón, en especial a Marcelo Pérez Medel quien tuvó paciencia conmigo, además de brindarme su amistad, así como a quienes indirectamente influyeron para la realización de éste trabajo: Dr. Enrique Daltabuit y M. en C. Leobardo Hdz.

ÍNDICE

INTRODUCCIÓN

Desde antes del surgimiento del sistema operativo Windows, los administradores no le daban suficiente importancia a la seguridad informática y no sólo en Windows si no también en otras plataformas, hoy en día es de suma importancia el considerar la seguridad como escencial, ya que cada vez más computadoras son atacadas y las pérdidas son cuantiosas, los sistemas UNIX actualmente son los que representan más confiabilidad y estabilidad que otros. Es por eso que se están creando cada día, mejores herramientas para proteger los sistemas de cómputo y así la integridad de los mismos. En este trabajo vamos a conocer de manera general el sistema operativo LINUX, así como relatar de manera clara y sencilla varios de los aspectos que intervienen para la detección de intrusos:

En el capítulo 1, se comenzará dando una introducción a lo que es la seguridad en cómputo; la cual inicía desde el lugar físico donde instalamos computadoras, servidores, ruteadores entre otros, además de conocer los diferentes intrusos que pueden entrar a un sistema con o sin autorización.

En el capítulo 2, muestra conceptos básicos que nos servirán para comprender el corazón de este trabajo, nos daremos cuenta de las características, ventajas y funcionamiento de LINUX, que es el sistema operativo donde analizaremos a los intrusos.

En el capítulo 3, se muestran los Sistemas de Detección de Intrusos como mecanismos encargados de localizar posibles ataques, así como clasificación y arquitectura.

En el capítulo 4, muestra la instalación y configuración de un Sistema de Detección de Intrusos, así como también se ejemplifica un caso real donde un servidor de la UNAM fue víctima de un hacker.

Se espera que la investigación en el área de la seguridad informática siga cobrando terreno para cubrir día con día los fallos que se tengan, cabe mencionar que en ésta institución solo se tiene una materia optativa que hace referencia a cuestiones de seguridad informática. Con el desarrollo de este trabajo se pretende;

Fomentar el interés entre los alumnos de la carrera de Ingeniería en Computación.

Describir los tipos de intrusos que pueden acceder a un sistema; así como también, detectar y conocer el comportamiento que tienen al violar un sistema de cómputo y en especial en el sistema operativo Linux.

Dar a conocer medidas de prevención; con el fin de atenuar las intrusiones informáticas, tanto en el laboratorio de cómputo del centro tecnológico Aragón, como en alguna otra entidad.

Capítulo I
INTRODUCCIÓN A LA SEGURIDAD EN CÓMPUTO.

Hoy en día la computadora y en sí la tecnología de la red está siendo muy popular, de hecho esas tecnologías han llegado a ser parte de nuestra vida cotidiana, por lo menos cada mes se sabe de noticias sobre una red que ha sufrido una intrusión en sus computadoras.

1.1 ¿QUÉ ES SEGURIDAD?

Se puede entender como seguridad; una característica de cualquier sistema que nos indica que ese sistema está libre de todo peligro, daño o riesgo, y que es, en cierta manera, infalible.

Mantener un sistema seguro (o fiable), consiste básicamente en garantizar tres aspectos: confidencialidad, integridad y disponibilidad. Algunos estudios integran la seguridad dentro de una propiedad más general de los sistemas, la confiabilidad; es entendida como el nivel de calidad del servicio ofrecido, se considera la disponibilidad como un aspecto al mismo nivel que la seguridad y no como parte de ella, por lo que dividen esta última en sólo las dos facetas: confidencialidad e integridad. En este trabajo no seguiremos esa corriente por considerarla minoritaria.

La confidencialidad nos dice que los objetos de un sistema han de ser accedidos únicamente por elementos autorizados a ello, y que esos elementos autorizados no van a convertir esa información en disponible para otras entidades; la integridad significa que los objetos sólo pueden ser modificados por elementos autorizados, y de una manera controlada, y la disponibilidad indica que los objetos del sistema tienen que permanecer accesibles a elementos autorizados; es el contrario de la negación de servicio. Generalmente tienen que existir los tres aspectos descritos para que haya seguridad: un sistema UNIX puede conseguir confidencialidad para un determinado

fichero haciendo que ningún usuario (ni siquiera el root) pueda leerlo, pero este mecanismo no proporciona disponibilidad alguna.

Dependiendo del entorno en que un sistema UNIX trabaje, a sus responsables les interesaría dar prioridad a un cierto aspecto de la seguridad. Por ejemplo, en un sistema militar se antepondría la confidencialidad de los datos almacenados o transmitidos sobre su disponibilidad: seguramente, es preferible que alguien borre información secreta, a que ese mismo atacante pueda leerla, o a que esa información esté disponible en un momento dado para los usuarios autorizados. En cambio, en un servidor NFS de un departamento se premiaría la disponibilidad frente a la confidencialidad: importa poco que un atacante lea una unidad, pero que esa misma unidad no sea leída por usuarios autorizados va a suponer una pérdida de tiempo y dinero. En un entorno bancario, la faceta que más ha de preocupar a los responsables del sistema es la integridad de los datos, frente a su disponibilidad o su confidencialidad: es menos grave que un usuario consiga leer el saldo de otro que el hecho de que ese usuario pueda modificarlo.

1.2 ¿QUÉ QUEREMOS PROTEGER?

Los tres elementos principales a proteger en cualquier sistema informático son el software, el hardware y los datos. Por hardware entendemos el conjunto formado por todos los elementos físicos de un sistema informático, como CPUs, terminales, cableado, medios de almacenamiento secundario (cintas, CD-ROMs, diskettes. . .) o tarjetas de red. Por software entendemos el conjunto de programas lógicos que hacen funcional al hardware, tanto sistemas operativos como aplicaciones, y por datos el conjunto de información lógica que manejan el software y el hardware, por ejemplo paquetes que circulan por un cable de red o entradas de una base de datos. Aunque generalmente en las auditorías de seguridad se habla de un cuarto elemento a proteger, los fungibles (elementos que se gastan o desgastan con el uso continuo, como papel de impresora, toners, cintas magnéticas, disquetes.), aquí no consideraremos la seguridad de estos elementos por ser externos al sistema UNIX. Habitualmente los datos constituyen el principal elemento de los tres a proteger, ya que es el

más amenazado y seguramente el más difícil de recuperar: con toda seguridad una máquina UNIX está ubicada en un lugar de acceso físico restringido, o al menos controlado, y además en caso de pérdida de una aplicación (o un programa de sistema, o el propio núcleo de UNIX) este software se puede restaurar sin problemas desde su medio original (por ejemplo, el CD-ROM con el sistema operativo que se utilizó para su instalación). Sin embargo, en caso de pérdida de una base de datos o de un proyecto de un usuario, no tenemos un medio "original" desde el que restaurar: hemos de pasar obligatoriamente por un sistema de copias de seguridad, y a menos que la política de copias sea muy estricta, es difícil devolver los datos al estado en que se encontraban antes de la pérdida.

Contra cualquiera de los tres elementos descritos anteriormente (pero principalmente sobre los datos) se pueden realizar multitud de ataques o, dicho de otra forma, están expuestos a diferentes amenazas. Generalmente, la taxonomía más elemental de estas amenazas las divide en cuatro grandes grupos: interrupción, interceptación, modificación y fabricación. Un ataque es clásico como interrupción si hace que un objeto del sistema se pierda, quede inutilizable o no disponible. Se tratará de una interceptación si un elemento no autorizado consigue un acceso a un determinado objeto del sistema, y de una modificación si además de conseguir el acceso consigue modificar el objeto; algunos autores consideran un caso especial de la modificación: la destrucción, entendiéndola como una modificación que inutiliza al objeto afectado. Por último, se dice que un ataque es una fabricación si se trata de una modificación destinada a conseguir un objeto similar al atacado de forma que sea difícil distinguir entre el objeto original y el "fabricado".

1.3 ¿DE QUÉ NOS QUEREMOS PROTEGER?

Se suele identificar a los atacantes únicamente como personas; esto tiene sentido si hablamos por ejemplo de responsabilidades por un delito informático. Pero en este trabajo es preferible hablar de "elementos" y no de personas: aunque a veces lo olvidemos, nuestro sistema puede verse perjudicado por múltiples entidades aparte de humanos, como por ejemplo programas, catástrofes naturales; si un usuario pierde un trabajo importante a causa de un ataque, poco le importaría que haya sido un intruso, gusano, incluso un simple error del administrador. A continuación se presenta una relación de los elementos que potencialmente pueden amenazar a nuestro sistema:

PERSONAS

No podemos engañarnos: la mayoría de ataques a nuestro sistema van a provenir en última instancia de personas que, intencionadamente o no , pueden causarnos enormes pérdidas. Generalmente se trataría de piratas que intentan conseguir el máximo nivel de privilegio posible aprovechando alguno (o algunos) de los riesgos lógicos de los que se hablará más adelante, especialmente agujeros del software.

PERSONAL

Las amenazas a la seguridad de un sistema provenientes del personal de la propia organización rara vez son tomadas en cuenta; se presupone un entorno de confianza donde a veces no existe, por lo que se pasa por alto el hecho de que casi cualquier persona de la organización, incluso el personal ajeno a la infraestructura informática (secretariado, personal de seguridad, personal de limpieza y mantenimiento. . .) puede comprometer la seguridad de los equipos.

Aunque los ataques pueden ser intencionados (en cuyo caso sus efectos son extremadamente dañinos, recordemos que nadie mejor que el propio personal de la organización conoce mejor los sistemas y sus debilidades), lo normal es que más que de ataques se trate de accidentes causados por un error o por desconocimiento de las normas básicas de seguridad: un empleado de

mantenimiento que corta el suministro eléctrico para hacer una reparación puede llegar a ser tan peligroso como el más experto de los administradores que se equivoca al teclear una orden y borra todos los sistemas de ficheros; y en el primer caso, el "atacante" ni siquiera ha de tener acceso lógico (ni físico) a los equipos, ni conocer nada sobre seguridad en Unix.

EX/EMPLEADOS

Otro gran grupo de personas potencialmente interesadas en atacar nuestro sistema son los antiguos empleados del mismo, especialmente los que no abandonaron el entorno por voluntad propia (y en el caso de redes de empresas, los que pasaron a la competencia). Generalmente, se trata de personas descontentas con la organización que pueden aprovechar debilidades de un sistema que conocen perfectamente para dañarlo como venganza por algún hecho que no consideran justo.

CURIOSOS

Junto con los crackers, los curiosos son los atacantes más habituales de sistemas Unix, recordemos que los equipos están trabajando en entornos donde se forma a futuros profesionales de la informática y las telecomunicaciones (gente que a priori tiene interés por las nuevas tecnologías), y recordemos también que las personas suelen ser curiosas por naturaleza; esta combinación produce una avalancha de estudiantes o personal intentando conseguir mayor privilegio del que tienen o intentando acceder a sistemas a los que oficialmente no tienen acceso. Y en la mayoría de ocasiones esto se hace simplemente para leer el correo de un amigo, enterarse de cuánto cobra un compañero, copiar un trabajo o comprobar que es posible romper la seguridad de un sistema concreto. Aunque en la mayoría de situaciones se trata de ataques no destructivos.

CRACKERS

Los crackers (crack=destruir) son aquellas personas que siempre buscan molestar a otros, piratear software protegido por leyes, destruir sistemas muy complejos mediante la transmisión de poderosos virus, etc. Esos son los crackers. Adolescentes inquietos que aprenden rápidamente este complejo oficio. Se diferencian con los Hackers porque no poseen ningún tipo de ideología cuando realizan sus "trabajos".

HACKERS

Los piratas ya no tienen un parche en su ojo ni un garfio en reemplazo de la mano. Llegando al año 2000, los piratas se presentan con un cerebro desarrollado, curioso y con muy pocas armas; una simple computadora y una línea telefónica, con esto los hackers pueden acceder a un sistema de cómputo. El término comenzó a usarse aplicándolo a un grupo de pioneros de la informática, a principios de la década de 1960. Desde entonces, y casi hasta finales de la década de 1970, un hacker era una persona obsesionada por conocer lo más posible sobre los sistemas informáticos. Pero a principios de la década de 1980, influenciados por la difusión de la película Juegos de Guerra, y el ampliamente publicado arresto de una "banda de hackers" conocida como la 414, los hackers pasaron a ser considerados como chicos jóvenes capaces de violar sistemas informáticos de grandes empresas y del gobierno.

Desgraciadamente, los medios de información y la comunidad científica social no ha puesto mucho esfuerzo por variar esta definición. El problema para llegar a una definición más precisa radica tanto en la poca información que hay, sobre sus actividades diarias, como en el hecho de que lo que se conoce de ellos no siempre cabe bajo las etiquetas de los delitos conocidos.

Es decir, no hay una definición legal que sea aplicable a los hackers, ni todas sus actividades conllevan la violación de las leyes. Esto lleva a que la aplicación del término varíe según los casos, dependiendo de los cargos que se puedan imputar y no a raíz de un claro de lo que significa ser un hacker, convierte a

esta en una etiqueta excesivamente utilizada para aplicar a muchos tipos de intrusiones informáticas.

Los términos, "hacker", "phreaker", "cracker" y "pirata" se presentan y definen tal y como los entienden aquellos que se identifican con estos papeles. Una palabra que aún no se encuentra en los diccionarios pero que ya suena en todas las personas que alguna vez se interesaron por la informática o leyeron algún diario. Proviene de "hack", el sonido que hacian los técnicos de las empresas telefónicas al golpear los aparatos para que funcionen.

Hoy es una palabra temida por empresarios y autoridades que desean controlar a quienes se divierten descifrando claves para ingresar a lugares prohibidos y tener acceso a información indebida. Sólo basta con repasar unas pocas estadísticas. Durante 1997, el 54 por ciento de las empresas norteamericanas sufrieron ataques de Hackers en sus sistemas.

La cultura popular define a los hackers como aquellos que, con ayuda de sus conocimientos informáticos consiguen acceder a los ordenadores de los bancos y de los negociados del gobierno. Bucean por información que no les pertenece, roban software caro y realizan transacciones de una cuenta bancaria a otra. Los criminólogos, por otra parte, describen a los hackers en términos menos halagadores. Hacen una clara distinción entre el hacker que realiza sus actividades por diversión y el empleado que de repente decide hacer algo malo. Por tanto, parece que tenemos una definición en la que caben dos extremos: por un lado, el moderno ladrón de bancos y por otro el inquieto. Ambas actividades (y todas las intermedias) son calificadas con el mismo término.

Contrariamente a lo que piensan los medios de comunicación, la mayoría de los hackers no destruyen y no dañan deliberadamente los datos. El hacerlo iría en contra de su intención de mezclarse con el usuario normal y atraería la atención sobre su presencia, haciendo que la cuenta usada sea borrada. Después de

gastar un tiempo sustancioso en conseguir la cuenta, el hacker pone una alta prioridad para que su uso no sea descubierto.

PHREAKERS

La cantidad de personas que se consideran phreakers, contrariamente a lo que sucede con los hackers, es relativamente pequeña. Pero aquellos que si se consideran phreakers lo hacen para explorar el sistema telefónico. La mayoría de la gente, aunque usa el teléfono, sabe muy poco acerca de él. Los phreakers, por otra parte, quieren aprender mucho sobre él. Sin embargo es, en estos últimos tiempos, cuando un buen Phreaker debe tener amplios conocimientos sobre informática, ya que la telefonía celular o el control de centrales es la parte primordial a tener en cuenta y/o emplean la informática para su procesado de datos.

TELEPIRATERÍA

Finalmente llegamos a la "telepiratería" del software. Consiste en la distribución ilegal de software protegido por los derechos de autor. No nos referiremos a la copia e intercambio de diskettes que se produce entre conocidos (que es igualmente ilegal), sino a la actividad que se realiza alrededor de los sistemas Warez[1] que se especializan en este tipo de tráfico.

El acceso a este tipo de servicios se consigue contribuyendo, a través de un módem telefónico, con una copia de un programa comercial. Este acto delictivo permite a los usuarios copiar, o "cargar", de tres a seis programas que otros hayan aportado. Así, por el precio de una sola llamada telefónica, uno puede amontonar una gran cantidad de paquetes de software.

En muchas ocasiones, incluso se evita pagar la llamada telefónica. Nótese que al contrario que las dos actividades de hacker y phreaker, no hay ninguna consideración al margen de "prestigio" o "motivación" en la telepiratería.

En este caso, el cometer los actos basta para "merecer" el título. La telepiratería está hecha para las masas. Al contrario de lo que sucede con los hackers y los

[1] Sitios en internet donde se encuentra software ilegal

phreakers, no requiere ninguna habilidad especial. Cualquiera que tenga un ordenador con módem y algún software dispone de los elementos necesarios para entrar en el mundo de la telepiratería. Debido a que la telepiratería no requiere conocimientos especiales, el papel de los piratas no inspira ningún tipo de admiración o prestigio en el submundo informático.

Un hacker mantiene la teoría de que son estos piratas los culpables de la mayoría de los fraudes con tarjetas de crédito telefónicas. "Los medios de comunicación afirman que son únicamente los hackers los responsables de las pérdidas de las grandes compañías de telecomunicaciones y de los servicios de larga distancia. Este no es el caso. Los hackers representan sólo una pequeña parte de estas pérdidas. El resto está causado por "los piratas" y ladrones que venden estos códigos en la calle.

Recuérdese que el objetivo de un hacker no es entrar en un sistema, sino aprender como funciona. El objetivo de un phreaker no es realizar llamadas de larga distancia gratis, sino descubrir lo que la compañía telefónica no explica sobre su red y el objetivo de un telepirata es obtener una copia del software más moderno para su ordenador.

Así, aunque un individuo tenga un conocimiento especial sobre los sistemas telefónicos, cuando realiza una llamada de larga distancia gratis para cargar un juego, está actuando como un telepirata. En cierto modo, esto es un puro argumento semántico. Independientemente de que a un hacker se le etiquete erróneamente como telepirata, los accesos ilegales y las copias no autorizadas de software comercial van a seguir produciéndose. Pero si queremos conocer los nuevos desarrollos de la era informática, debemos identificar y reconocer los tres tipos de actividades con que nos podemos encontrar. El agrupar los tres tipos bajo una sóla etiqueta es más que impreciso, ignora las relaciones funcionales y diferencias entre ellos. Hay que admitir, de

todas formas, que siempre habrá alguien que esté en desacuerdo con las diferencias que se han descrito entre los grupos.

En el desarrollo de esta investigación, los individuos que realizan actualmente estas actividades no se ponen de acuerdo en cuanto a donde están las fronteras. Las categorías y papeles, como se ha indicado previamente, no son mutuamente exclusivos. En particular, el mundo de los hackers y los phreakers están muy relacionados.

1.4 SEGURIDAD FÍSICA

La seguridad de un sistema de cómputo comienza desde la seguridad física que se tiene en el centro de operaciones computacional; la seguridad física de los sistemas informáticos consiste en la aplicación de barreras físicas y procedimientos de control como medidas de prevención y medidas contra las amenazas a los recursos y la información confidencial. Particularizando para el caso de equipos Unix y sus centros de operación, por seguridad física podemos entender todas aquellos mecanismos (generalmente de prevención y detección) destinados a proteger físicamente cualquier recurso del sistema; estos recursos son desde un simple teclado, hasta una copia de seguridad con toda la información que hay en el sistema, pasando por el CPU de la máquina. Desgraciadamente, la seguridad física es un aspecto olvidado con demasiada frecuencia a la hora de hablar de seguridad informática en general; en muchas organizaciones se suelen tomar medidas para prevenir o detectar accesos no autorizados o negaciones de servicio, pero rara vez para prevenir la acción de un atacante que intenta acceder físicamente a la sala de operaciones o al lugar donde se depositan las impresiones del sistema. Esto motiva que en determinadas situaciones un atacante se decline por aprovechar vulnerabilidades físicas en lugar de lógicas, ya que posiblemente le sea más fácil robar una imagen completa del sistema que intentar acceder a él, mediante fallos en el software.

Debemos ser conscientes que la seguridad física es demasiado importante como para ignorarla: un ladrón que roba un ordenador para venderlo, un incendio o un pirata que accede sin problemas a la sala de operaciones nos pueden hacer mucho más daño que un intruso que intenta conectar remotamente con una máquina no autorizada; no importa que utilicemos los más avanzados medios de cifrado para conectar a nuestros servidores, si no tenemos en cuenta factores físicos, estos esfuerzos para proteger nuestra información no van a servir de nada. Hemos de pensar que si un atacante puede llegar con total libertad hasta una estación puede por ejemplo abrir el CPU y llevarse un disco duro; sin necesidad de privilegios en el sistema, sin importar la robustez de nuestros cortafuegos, sin ni siquiera una clave de usuario, el atacante podría seguramente modificar la información almacenada, destruirla o simplemente leerla.

Una muestra de que la importancia en seguridad física en las computadoras no se ha hecho del ámbito público, es que día a día personas sin conocimientos en computación, roban los servidores; tal es el caso de dos personas que entraron en el centro de cómputo y gestión de cargamentos de las aduanas del Aeropuerto Internacional de Sydney y robaran físicamente dos *mainframes* .

A veces, la realidad supera la ficción y los mejores delitos resultan ser los más descarados y obvios. Tras dar nombres y firmas falsas, se les permitió el acceso a la sala de datos, que no contaba con personal de vigilancia. En México, también sucede lo mismo, frecuentemente en el Instituto Mexicano del Seguro Social, el robo del equipo de cómputo se ha convertido en un problema serio. Y aunque no se obtengan ganancias monetarias, se cuestiona el hecho de que cualquier persona pueda tener acceso a las computadoras y por consecuente a la información.

Según un reciente estudio del Computer Security Institute (CSI), el 75% de 563 empresas, instituciones y universidades norteamericanas encuestadas perdieron dinero, en 1997, por culpa de fallos de seguridad en sus sistemas.

Aunque de Internet sólo provienen el 20 por ciento de los ataques informáticos sufridos por las empresas la mayoría se perpetran desde dentro. Mientras, el peligro crece, dicen las estadísticas: en 1988 el Computer Emergency Response Team (CERT) norteamericano registro los siguientes datos. Ataques a empresas estadounidenses en la red, conocidos por el CERT;

INCIDENTES AÑO 2002

Acceso a cuentas privilegiadas.... 96
Escaneo de puertos 706
Denegación de Servicio 23
Troyanos y gusanos 622
Uso no autorizado de "proxies" 22
Otros 26
TOTAL: 1495

AÑO 2001
Escanos o intentos de entrada..... 21
Accesos ilegales a máquinas..... 28
Denegaciones de servicio (DoS)..... 16
Infección gusanos/virus..... 118
Correo basura..... 4
Otros..... 3

TOTAL: 190

AÑO 2000

Escaneos o intentos de entrada... 268
Accesos ilegales a máquinas..... 61
Correo basura..... 74
Denegaciones de servicio..... 24
Troyanos..... 14
Otros..... 46

TOTAL: 490

AÑO 1999

Escaneos o intentos de entrada..... 30
Accesos ilegales y denegaciones... 38
Correo basura.....78
Otros.....94

TOTAL: 240

Los datos antes citados son de acuerdo a lo reportado por CERT, pero cabe mencionar que en realidad el número de ataques realmente es mayor, se puede ver que día a día la cantidad de atacantes va incrementándose.

Los terremotos son desastres naturales que pueden pasar en cualquier parte de este planeta, claro que influye la localización geográfica: no nos encontramos en una zona donde se suelan producir temblores de intensidad considerable; como Japón o Estados Unidos. La probabilidad de un temblor esta presente. Los terremotos no suelen alcanzar la magnitud necesaria para causar daños en los equipos. Por tanto, no se suelen tomar medidas serias contra los movimientos sísmicos, no vale la pena invertir dinero para minimizar sus efectos, sin embargo, son considerados en la parte de seguridad física.

1.5 LA INTRODUCCIÓN DEL DERECHO SOBRE LOS DELITOS INFORMÁTICOS

En la actualidad las computadoras se utilizan no sólo como herramientas auxiliares de apoyo a diferentes actividades humanas, sino como medio eficaz para obtener y conseguir información, lo que las ubica también como un nuevo medio de comunicación, y condiciona su desarrollo a la informática; tecnología cuya esencia se resume en la creación, procesamiento, almacenamiento y transmisión de datos.

La informática está hoy presente en casi todos los campos de la vida moderna, es algo que a la gente dedicada al cómputo nos gusta, en lo particular quise hacer mención sobre legislación en delitos informáticos, porque me llama la atención la rama del derecho. Con mayor o menor rapidez todas las ramas del saber humano se rinden ante los progresos tecnológicos, y comienzan a utilizar los sistemas de Información para ejecutar tareas que en otros tiempos realizaban manualmente.

El progreso cada día más importante y sostenido de los sistemas computacionales permite hoy procesar y poner a disposición de la sociedad una cantidad creciente de información de toda naturaleza, al alcance concreto de millones de interesados y de usuarios. Las más diversas esferas del conocimiento humano, en lo científico, en lo técnico, en lo profesional y en lo

personal están siendo incorporados a sistemas informáticos que, en la práctica cotidiana, de hecho sin limitaciones, entrega con facilidad a quien lo desee un conjunto de datos que hasta hace unos años sólo podían ubicarse luego de largas búsquedas y selecciones en que el hombre jugaba un papel determinante y las máquinas existentes tenian el rango de equipos auxiliares para imprimir los resultados. En la actualidad, en cambio, ese enorme caudal de conocimiento puede obtenerse, además, en segundos o minutos, transmitirse incluso documentalmente y llegar al receptor mediante sistemas sencillos de operar, confiables y capaces de responder casi toda la gama de interrogantes que se planteen a los archivos informáticos.

Puede sostenerse que hoy las perspectivas de la informática no tienen límites previsibles y que aumentan en forma que aún puede impresionar a muchos actores del proceso. Este es el panorama de este nuevo fenómeno científico-tecnológico en las sociedades modernas. Por ello, llega a sostenerse que la Informática es hoy, una forma de "poder social". Las facultades que el fenómeno pone a disposición de gobiernos y de particulares, con rapidez y ahorro consiguiente de tiempo y energía, configuran un cuadro de realidades de aplicación y de posibilidades de juegos lícito e ilícito, en donde es necesario el derecho para regular los múltiples efectos de una situación, nueva y de tantas potencialidades en el medio social.

Los progresos mundiales de las computadoras, el creciente aumento de las capacidades de almacenamiento y procesamiento, la miniaturización de los chips de las computadoras instalados en productos industriales, la fusión del proceso de la información con las nuevas tecnologías de comunicación, así como la investigación en el campo de la inteligencia artificial, ejemplifican el desarrollo actual definido a menudo como la "era de la información".

Esta marcha de las aplicaciones de la informática no sólo tiene un lado ventajoso sino que plantea también problemas de significativa importancia para

el funcionamiento y la seguridad de los sistemas informáticos en los negocios, la administración, la defensa y la sociedad. Debido a esta vinculación, el aumento del nivel de los delitos relacionados con los sistemas informáticos registrados en la última década en los Estados Unidos, Europa Occidental, Australia y Japón, representa una amenaza para la economía de un país y también para la sociedad en su conjunto.

De acuerdo con la definición elaborada por un grupo de expertos[2], en mayo de 1983, el término delitos relacionados con las computadoras se define como cualquier comportamiento antijurídico, no ético o no autorizado, relacionado con el proceso automático de datos y/o transmisiones de datos. La amplitud de este concepto es ventajosa, puesto que permite el uso de las mismas hipótesis de trabajo para toda clase de estudios penales, criminológicos, económicos, preventivos o legales.

En la actualidad la tecnología de la información se ha implantado en casi todos los países. Tanto en la organización y administración de empresas y administraciones públicas como en la investigación científica, en la producción industrial o en el estudio e incluso en el ocio, el uso de la informática es en ocasiones indispensable y hasta conveniente. Sin embargo, junto a las incuestionables ventajas que presenta comienzan a surgir algunas facetas negativas, como por ejemplo, lo que ya se conoce como "criminalidad informática".

El espectacular desarrollo de la tecnología informática ha abierto las puertas a nuevas posibilidades de delincuencia antes impensables. La manipulación fraudulenta de los ordenadores con ánimo de lucro, la destrucción de programas o datos, el acceso y la utilización indebida de la información que puede afectar la esfera de la privacidad, son algunos de los procedimientos relacionados con el procesamiento electrónico de datos mediante los cuales es

[2] Personal participante en el manual de naciones unidas para el control de delitos informáticos

posible obtener grandes beneficios económicos o causar importantes daños materiales o morales. Pero no sólo la cuantía de los perjuicios así ocasionados es a menudo infinitamente superior a la que es usual en la delincuencia tradicional, sino que también son mucho más elevadas las posibilidades de que no lleguen a descubrirse. Se trata de una delincuencia de especialistas capaces muchas veces de borrar toda huella de los hechos. En este sentido, la informática puede ser el objeto del ataque o el medio para cometer otros delitos. La informática reúne unas características que la convierten en un medio idóneo para la comisión de muy distintas modalidades delictivas, en especial de carácter patrimonial (estafas, apropiaciones indebidas, etc.).

La idoneidad proviene, básicamente, de la gran cantidad de datos que se acumulan, con la consiguiente facilidad de acceso a ellos y la relativamente fácil manipulación de esos datos. La importancia reciente de los sistemas de datos, por su gran incidencia en la marcha de las empresas, tanto públicas como privadas, los ha transformado en un objeto cuyo ataque provoca un perjuicio enorme, que va mucho más allá del valor material de los objetos destruidos. A ello se une que estos ataques son relativamente fáciles de realizar, con resultados altamente satisfactorios y al mismo tiempo procuran a los autores una probabilidad bastante alta de alcanzar los objetivos sin ser descubiertos.

El estudio de los distintos métodos de destrucción y/o violación del hardware y el software es necesario para determinar cuál será la dirección que deberá seguir la protección jurídica de los sistemas informáticos, ya que sólo conociendo el mecanismo de estos métodos es posible encontrar las similitudes y diferencias que existen entre ellos. En consecuencia, la legislación sobre protección de los sistemas informáticos ha de perseguir acercarse lo más posible a los distintos medios de protección ya existentes, creando una nueva regulación sólo en aquellos aspectos en los que, en base a las peculiaridades del objeto de protección, sea imprescindible. Si se tiene en cuenta que los sistemas informáticos, pueden entregar datos e informaciones sobre miles de

personas, en aspectos tan fundamentales para el normal desarrollo y funcionamiento de diversas actividades; bancarias, financieras, tributarias, previsionales y de identificación de las personas. Y si a ello se agrega que existen bancos de datos, empresas o entidades dedicadas a proporcionar, cualquier información, sea de carácter personal o sobre materias de las más diversas disciplinas a un Estado o particulares; se comprenderá que están en juego o podrían a llegar a estarlo de modo dramático, algunos valores colectivos y los consiguientes bienes jurídicos que el ordenamiento jurídico-institucional debe proteger.

No es la amenaza potencial de la computadora sobre el individuo lo que provoca desvelo, sino la utilización real por el hombre de los sistemas de información con fines de espionaje. No son los grandes sistemas de información los que afectan la vida privada sino la manipulación o el consentimiento de ello, por parte de individuos poco conscientes e irresponsables de los datos que dichos sistemas contienen.

La protección de los sistemas informáticos puede abordarse tanto desde una perspectiva penal como de una perspectiva civil o comercial, e incluso de derecho administrativo. Estas distintas medidas de protección no tienen porque ser excluyentes unas de otras, sino que, por el contrario, éstas deben estar estrechamente vinculadas.

Por eso, dadas las características de esta problemática sólo a través de una protección global, desde los distintos sectores del ordenamiento jurídico, es posible alcanzar una cierta eficacia en la defensa de los ataques a los sistemas informáticos.

PROPUESTAS PARA FORMAR UNA LEGISLACIÒN INFORMÁTICA EN MÉXICO Y OTROS PAISES.

Un análisis de las legislaciones que se han promulgado en diversos países arroja que las normas jurídicas puestas en vigor están dirigidas a proteger la utilización abusiva de la información reunida y procesada mediante el uso de computadoras, e incluso en algunas de ellas se ha previsto formar órganos especializados que protejan los derechos de los ciudadanos amenazados por los ordenadores.

Desde hace aproximadamente diez años la mayoría de los países europeos han hecho todo lo posible para incluir dentro de la ley, la conducta punible penalmente, como el acceso ilegal a sistemas de cómputo o el mantenimiento ilegal de tales accesos, la difusión de virus o la intercepción de mensajes informáticos.

En la mayoría de las naciones occidentales existen normas similares a los países europeos. Todos estos enfoques están inspirados por la misma preocupación de contar con comunicaciones electrónicas, transacciones e intercambios tan confiables y seguros como sea posible.

Dar un concepto sobre delitos informáticos no es una labor fácil y esto en razón de que su misma denominación alude a una situación muy especial, ya que para hablar de "delitos" en el sentido de acciones tipificadas o contempladas en textos jurídico-penales, se requiere que la expresión "delitos informáticos" esté consignada en los códigos penales, lo cual en nuestro país, al igual que en muchos otros, no ha sido objeto de tipificación aún; sin embargo, algunos especialistas en derecho informático, entidades públicas y privadas están dando prouestas para que el gobierno imponga una legislación informática. A continuación se mostrará una propuesta para la tipificación sobre delitos informáticos, hecha por un senador de un partido político, denominada de la siguiente manera; "Ley Federal para la protección de datos personales".

En esta propuesta se comprenden 5 capítulos, el primero relativo a las disposiciones generales, el segundo a los derechos de los interesados o titulares de los datos, así como a los responsables de los registros, el tercero al Instituto Federal de Protección de Datos Personales, el cuarto a las sanciones, y el quinto a la acción protectora de datos.

En el Capítulo 1 se mencionan los objetos de la ley, las expresiones equivalentes, el ámbito de validez, y los principios sobre los cuales descansa la ley, sobresaliendo la circunstancia de que en ningún caso se pueden afectar los archivos, registros, bases o bancos de datos ni las fuentes de información periodísticas.

En el Capítulo 2, se regulan los derechos de los interesados, estableciendo un catálogo de obligaciones correspondientes a los organismos públicos y privados titulares de los datos.

Conviene resaltar los derechos de los interesados para solicitar al Instituto Federal de Protección de Datos Personales la existencia de registros personales, las finalidades y la identidad de los responsables, así como el derecho de pedir a los responsables de archivos, registros, bases o bancos de datos informes, y de pedir de igual manera la inclusión, actualización, complementación, rectificación, suspensión y cancelación de los registros de datos que les correspondan, siempre que no se lesionen derechos de terceros o se atente contra intereses de carácter general o social.

Se habla de las responsabilidades de los titulares de los Registros de las diversas categorías (públicos y privados) de bases o bancos de datos, quienes deben adoptar todas las medidas necesarias para la seguridad y conservación idónea de los datos, imponiéndoseles, incluso, el deber de secreto que se extiende a todos aquellos que hayan intervenido en el tratamiento automatizado de los datos.

En el Capítulo 3, se establecen las líneas generales para la creación y operación del organismo que tendrá por objeto el control de los responsables de los registros, bases o bancos de datos, así como sus atribuciones, en las que destaca la facultad de sancionar a los responsables de los archivos o registros por la comisión de violaciones leves y graves a esta ley.

En el Capítulo 4, se propone la regulación específica de las sanciones, que van desde el apercibimiento hasta la cancelación de los registros, archivos, bases o bancos de datos.

En el Capítulo 5, se propone la regulación de un procedimiento especial del que conozcan los juzgados de distrito competentes con relación a causas federales, pues ello persigue que se resuelvan las controversias de manera pronta y sin obstáculos en tiempos más breves que los que corresponden, incluso a los juicios de amparo, pues una administración de justicia que no se otorgue en esos términos, prácticamente inutilizaría el recurso.

Finalmente, en las disposiciones transitorias se prevé que los archivos, bases o bancos de datos existentes se puedan registrarse, conforme lo determine el reglamento, en un lapso posterior al establecimiento del Instituto Federal de Protección de Datos.

Por su parte, el tratadista penal italiano Carlos Sarzana[3], sostiene que los delitos informáticos son "cualquier comportamiento criminal en que la computadora está involucrada como material, objeto o mero símbolo".

El autor mexicano Julio Tellez Váldez[4] señala que los delitos informáticos son "actitudes ilícitas en que se tienen a las computadoras como instrumento o fin (concepto atípico) o las conductas típicas, antijurídicas y culpables en que se tienen a las computadoras como instrumento o fin (concepto típico)". Según

[3] Abogado penalista interesado en la legislación informática en méxico
[4] www.fiadi.org/html/curriculum/curriculum-j_tellez.htm

Tellez Valdez, este tipo de acciones presentan las siguientes caracteristicas principales: A continuación se mencionaran algunos puntos de la propuesta;

- Son conductas criminales de cuello blanco (white collar crime), en tanto que sólo un determinado número de personas con ciertos conocimientos (en este caso técnicos) pueden llegar a cometerlas.
- Son acciones ocupacionales, en cuanto a que muchas veces se realizan cuando el sujeto se halla trabajando.
- Son acciones de oportunidad, ya que se aprovecha una ocasión creada o altamente intensificada en el mundo de funciones y organizaciones del sistema tecnológico y económico.
- Provocan serias pérdidas económicas, ya que casi siempre producen "beneficios" de más de cinco cifras a aquellos que las realizan.
- Ofrecen posibilidades de tiempo y espacio, ya que en milésimas de segundo y sin una necesaria presencia física pueden llegar a consumarse.

Son muchos los casos y pocas las denuncias, y todo ello debido a la misma falta de regulación por parte del Derecho. Son muy sofisticados y relativamente frecuentes en el ámbito militar, presentan grandes dificultades para su comprobación; esto por su mismo carácter técnico. En su mayoría son imprudenciales y no necesariamente se cometen con intención. Además ofrecen facilidades para su comisión a los menores de edad. Tienden a proliferar cada vez más, por lo que requieren una urgente regulación. Por el momento siguen siendo ilícitos impunes de manera manifiesta ante la ley.

Asimismo, este autor clasifica a estos delitos, de acuerdo a dos criterios:

- El primer criterio es considerando como instrumento o medio.

En esta categoría se encuentran las conductas criminales que se valen de las computadoras como método, medio o símbolo en la comisión del ilícito, por ejemplo:

- Falsificación de documentos vía computarizada (tarjetas de crédito, cheques, etc.).
- Variación de los activos y pasivos en la situación contable de las empresas.
- Planeamiento y simulación de delitos convencionales (robo, homicidio, fraude, etc.)
- Lectura, sustracción o copiado de información confidencial.

- El segundo criterio considera la modificación de datos tanto en la entrada como en la salida.

- Aprovechamiento indebido o violación de un código para penetrar a un sistema introduciendo instrucciones inapropiadas.
- Variación en cuanto al destino de pequeñas cantidades de dinero hacia una cuenta bancaria apócrifa.
- Uso no autorizado de programas de cómputo.
- Introducción de instrucciones que provocan "interrupciones" en la lógica interna de los programas.
- Alteración en el funcionamiento de los sistemas, a través de los virus informáticos.
- Obtención de información residual impresa en papel luego de la ejecución de trabajos.
- Acceso a áreas informatizadas en forma no autorizada.
- Intervención en las líneas de comunicación de datos o teleproceso.

En esta categoría, se enmarcan las conductas criminales que van dirigidas contra las computadoras, accesorios o programas como entidad física, como por ejemplo:

- Programación de instrucciones que producen un bloqueo total al sistema.
- Destrucción de programas por cualquier método.
- Daño a la memoria.
- Atentado físico contra la máquina o sus accesorios.
- Sabotaje político o terrorismo en que se destruya o surja un apoderamiento de los centros neurálgicos computarizados.

Por otra parte, existen diversos tipos de delito que pueden ser cometidos y que se encuentran ligados directamente a acciones efectuadas contra los propios sistemas como son:

Acceso no autorizado: Uso ilegitimo de passwords y la entrada a un sistema informático sin la autorización del propietario.

Destrucción de datos: Los daños causados en la red mediante la introducción de virus, bombas lógicas, etc.

Infracción al copyright de bases de datos: Uso no autorizado de información almacenada en una base de datos.

Interceptación de e-mail: : Lectura de un mensaje electrónico ajeno.

Estafas electrónicas: A través de compras realizadas haciendo uso de la red.

Transferencias de fondos: Engaños en la realización de este tipo de transacciones.

Por otro lado, la red Internet permite dar soporte para la comisión de otro tipo de delitos:

Espionaje: Acceso no autorizado a sistemas informáticos gubernamentales y de grandes empresas e interceptación de correos electrónicos.

Terrorismo: Mensajes anónimos aprovechados por grupos terroristas para remitirse consignas y planes de actuación a nivel internacional.

Narcotráfico: Transmisión de fórmulas para la fabricación de estupefacientes, para el blanqueo de dinero y para la coordinación de entregas y recogidas.

Otros delitos: Las mismas ventajas que encuentran en la Internet los narcotraficantes pueden ser aprovechadas para la planificación de otros delitos como el tráfico de armas, proselitismo de sectas, propaganda de grupos

extremistas, y cualquier otro delito que pueda ser trasladado de la vida real al ciberespacio o al revés.

Existen varias propuestas para la legislación informática en México; Actualmente existe en México la Asociación Nacional de investigadores en informática jurídica[5] A.C., autores, IPN, UNAM, BBVA-Bancomer, partidos políticos y demás gente interesada en la materia, pero cabe resaltar que el estado es quién tiene la palabra.

SITUACIÓN EN EL RESTO DEL MUNDO.
En la Argentina, aún no existe legislación específica sobre los llamados delitos informáticos. Sólo están protegidas las obras de bases de datos y de software, agregados a la lista de ítems contemplados por la Ley de propiedad intelectual. En dicho Decreto se definen:
A) Obras de software: Las producciones que se ajusten a las siguientes definiciones:
Los diseños, tanto generales como detallados, del flujo lógico de los datos en un sistema de computación.

B) Los programas de computadoras, tanto en versión "fuente", principalmente destinada al lector humano, como en su versión "objeto", principalmente destinada a ser ejecutada por la computadora.

C) La documentación técnica, con fines tales como explicación, soporte o entrenamiento, para el desarrollo, uso o mantenimiento de software.

D) Obras de base de datos: Se las incluye en la categoría de "obras literarias", y el término define a las producciones "constituidas por un conjunto organizado de datos interrelacionados, compilado con miras a su almacenamiento, procesamiento y recuperación mediante técnicas y sistemas informáticos".

[5] Benjamin Franklin 50, letra I,Col. Escandon, 11800 Mexico, D.F

De acuerdo con los códigos vigentes, para que exista robo o hurto debe afectarse una cosa, entendiendo como cosas aquellos objetos materiales susceptibles de tener algún valor, la energía y las fuerzas naturales susceptibles de apropiación. Asimismo, la situación legal ante daños infligidos a la información es problemática.

Un artículo del Código Civil argentino declara *"el acto ilícito ejecutado a sabiendas y con intención de dañar la persona o los derechos del otro se llama, en este Código, delito*[6], obligando a reparar los daños causados por tales delitos.

En caso de probarse la existencia de delito de daño por destrucción de la cosa ajena, *"la indemnización consistirá en el pago de la cosa destruida; si la destrucción de la cosa fuera parcial, la indemnización consistirá en el pago de la diferencia de su valor y el valor primitivo"* .

Existe la posibilidad de reclamar indemnización cuando el hecho no pudiera ser considerado delictivo, en los casos en que *"alguien por su culpa o negligencia ocasiona un daño a otro"*.

En todos los casos, el resarcimiento de daños consistirá en la reposición de las cosas a su estado anterior, excepto si fuera imposible, en cuyo caso la indemnización se fijará en dinero" .

El mayor inconveniente es que no hay forma de determinar fehacientemente cuál era el estado anterior de los datos, puesto que la información en estado digital es fácilmente adulterable. Por otro lado, aunque fuera posible determinar el estado anterior, sería difícil determinar el valor que dicha información tenía,

[6] www.derechoargentino.com.ar codigo_civil.htm

pues es sabido que el valor de la información es subjetivo, es decir, que depende de cada uno y del contexto.

Lo importante en este tema es determinar que por más que se aplique la sanción, la misma resulta insuficiente a efectos de proteger los programas de computación, los sistemas o la información en ellos contenidos de ciertas conductas delictivas tales como: el ingreso no autorizado, la violación de secretos, el espionaje, el uso indebido, el sabotaje, etc.
No obstante, existen en el Congreso Nacional diversos proyectos de ley que contemplan esta temática; aunque sólo dos de ellos cuentan actualmente con estado parlamentario.

ALEMANIA.
Este país sancionó en 1986 la Ley contra la Criminalidad Económica, que contempla los siguientes delitos:

- Espionaje de datos.
- Estafa informática.
- Alteración de datos.
- Sabotaje informático.

AUSTRIA.
La Ley de reforma del Código Penal, del 22 de diciembre de 1987, sanciona a aquellos que con dolo causen un perjuicio patrimonial a un tercero influyendo en el resultado de una elaboración de datos automática a través de la confección del programa, por la introducción, cancelación o alteración de datos o por actuar sobre el curso del procesamiento de datos. Además, contempla sanciones para quienes cometen este hecho utilizando su profesión de especialistas en sistemas.

GRAN BRETAÑA.

Debido a un caso de hacking en 1991, comenzó a regir en este país la Computer Misuse Act (Ley de Abusos Informáticos). Mediante esta ley el intento, exitoso o nó, de alterar datos informáticos es penado con hasta cinco años de prisión o multas.

Esta ley tiene un apartado que especifica la modificación de datos sin autorización. Los virus están incluidos en esa categoría. El liberar un virus tiene penas desde un mes a cinco años, dependiendo del daño que causen.

HOLANDA.

El 1° de Marzo de 1993 entró en vigencia la Ley de Delitos Informáticos, en la cual se penaliza el hacking, el preacking (utilización de servicios de telecomunicaciones evitando el pago total o parcial de dicho servicio), la ingeniería social (arte de convencer a la gente de entregar información que en circunstancias normales no entregaría), y la distribución de virus. La distribución de virus está penada de distinta forma si se escaparon por error o si fueron liberados para causar daño.

Si se demuestra que el virus se escapó por error, la pena no superará el mes de prisión; pero, si se comprueba que fueron liberados con la intención de causar daño, la pena puede llegar hasta los cuatro años de prisión.

FRANCIA.

En enero de 1988, este país dictó la Ley relativa[7] al fraude informático, la cual prevé penas de dos meses a dos años de prisión y multas de diez mil a cien mil francos por la intromisión fraudulenta que suprima o modifique datos.

Asimismo, esta ley establece en su artículo 462-3 una conducta intencional y a sabiendas de estar vulnerando los derechos de terceros que haya impedido o alterado el funcionamiento de un sistema de procesamiento automatizado de

[7] www.zimmer.csufresno.edu ~haralds legal foreign codigos.htm

datos. Por su parte el artículo 462-4 también incluye en su tipo penal una conducta intencional y a sabiendas de estar vulnerando los derechos de terceros, en forma directa o indirecta, haya introducido datos en un sistema de procesamiento automatizado o haya suprimido modificado los datos que éste contiene, o sus modos de procesamiento o de transmisión.

También la legislación francesa establece un tipo doloso y pena el mero acceso, agravando la pena cuando resultare la supresión o modificación de datos contenidos en el sistema, o bien en la alteración del funcionamiento del sistema (sabotaje). Por último, esta ley en su artículo 462-2, sanciona tanto el acceso al sistema como al que se mantenga en él y aumenta la pena correspondiente si de ese acceso resulta la supresión o modificación de los datos contenidos en el sistema o resulta la alteración del funcionamiento del sistema.

ESPAÑA.

En tanto, el artículo (264-2) establece que se aplicará la pena de prisión de uno a tres años y multa a quien por cualquier medio destruya, altere, inutilice o de cualquier otro modo dañe los datos, programas o documentos electrónicos ajenos contenidos en redes, soportes o sistemas informáticos.

El nuevo Código Penal de España sanciona en forma detallada esta categoría delictual (Violación de secretos/Espionaje/Divulgación), aplicando pena de prisión y multa, agravándolas cuando existe una intensión dolosa y cuando el hecho es cometido por parte de funcionarios públicos se penaliza con inhabilitación.

En materia de estafas electrónicas, el nuevo Código Penal de España, en su artículo 248, sólo tipifica las estafas con ánimo de lucro valiéndose de alguna manipulación informática, sin detallar las penas a aplicar en el caso de la comisión del delito.

CHILE.

Chile fué el primer país latinoamericano en sancionar una Ley contra delitos informáticos, la cual entró en vigencia el 7 de junio de 1993.

Según esta ley, la destrucción o inutilización de los datos contenidos dentro de una computadora es castigada con penas desde un año y medio a cinco años de prisión. Asimismo, dentro de esas consideraciones se encuentran los virus.

Esta ley preveé en el Art. 1°, el tipo legal vigente de una conducta maliciosa tendiente a la destrucción o inutilización de un sistema de tratamiento de información o de sus partes componentes o que dicha conducta impida, obstaculice o modifique su funcionamiento.

Con esto, se ha mostrado un panorama sobre la seguridad en cómputo, cabe señalar que aún se tienen en México deficiencias en la tipificación de legislaciones informáticas y por consecuente; personas interesadas en la materia.

Capítulo II
SISTEMA OPERATIVO LINUX.

En el Capítulo anterior, se obtuvo el conocimiento necesario para comprender la seguridad en computación, ahora se hablará sobre Linux, el cuál es uno de los sistemas operativos importantes en la actualidad y de interés en el ámbito acádemico, así como también un sistema operativo tema de estudio en diferentes universidades, aunque también vulnerable.

2.1 ¿QUÉ ES UN SISTEMA OPERATIVO?

Un Sistema Operativo es un administrador. La función de cualquier sistema operativo es la de administrar los recursos del sistema (estos pueden ser disco duro, monitor, memoria) en el que se ejecuta y libra a los usuarios de comprender los detalles de hardware, es decir, funge como intermediario entre estos y el usuario.

2.2 HISTORIA DE LINUX

La historia de LINUX está muy ligada a UNIX, entonces para empezar, conoceremos un poco de la historia de este último.

Antes de los 50's las computadoras eran monousuario, ya que sólo una persona podía estar trabajando en ellas al mismo tiempo, más adelante, al comienzo de lo 60's aparecieron los sistemas de tiempo compartido, y gracias a ellos, varios usuarios podían estar conectados a la misma vez a una computadora. A mediados de los 60's las empresas MIT[8], AT&T y General Electric (compañías dedicadas a las comunicaciones)se juntaron para realizar un gran proyecto, se trataba de hacer un Sistema Operativo de gran potencia al que denominaron MULTICS.

[8] Instituto Tecnológico de Massachussets

El proyecto fué un fracaso pero uno de los programadores del MIT que habría trabajado en el proyecto, Ken Thompson, y un grupo de colaboradores decidieron escribir una versión miniatura de MULTICS.

Unos de los compañeros de Ken, Brian Kernigham, en una reunión de equipo bromeando llamó al sistema de Ken Thompson UNICS. UNICS fue un gran éxito y Ken decidió que UNIX era un nombre más atractivo que UNICS. Había nacido UNIX.

Un famoso artículo del año 1974 que describía UNIX atrajo la atención de las universidades que solicitaron el código fuente para estudiarlo y explicarlo en las aulas.

Muy pronto, UNIX logró una gran aceptación en la comunidad científica y el interés por este sistema operativo comenzó a extenderse. A partir de este momento comienza una verdadera avalancha de versiones del sistema, lo que primero en un principio empezó como un proyecto de investigación se convirtió más tarde en un gran negocio.

Las más importantes de todas las versiones de UNIX fueron la BSD, de la Universidad de California en Berkeley, que contenía una serie de mejoras que hicieron a UNIX un sistema operativo más amigable, y la System V. Esta última surgió de la fusión de las respectivas versiones de UNIX de AT&T Bell Laboratories, los creadores del sistema, y Sun Microsystems. A pesar del éxito comercial de UNIX y de su aceptación como sistema operativo, el código fuente de UNIX no podía ser explicado en aulas universitarias, de modo que el desarrollo de sistemas operativos volvía a ser una ciencia restringida a un reducido grupo de empresas y personas.

Ante esta situación, el profesor Andrew Tenembaum, de la Universidad de Vrije, en Amsterdam, decidió imitar a Ken Thompson cuando escribió el código de

UNIX basándose en MULTICS, e inspirándose en UNIX llevó a cabo un nuevo sistema operativo mucho mas reducido, al que llamó MINIX (de Mini-UNIX). MINIX había sido desarrollado en una IBM PC y, sin embargo ofrecía las mismas llamadas al sistema que UNIX V7. Tenembaum hizo público el código de MINIX, y su texto aún se usa en la mayoría de las universidades del planeta para enseñar las bases del diseño de sistemas operativos.

Linux[9] fué concebido por un finlandés de 21 años llamado Linus Torvalds estudiante de la universidad de Helsinki, quien comenzó trabajando sobre los fuente de Minix (un pequeño UNIX desarrollado por Andy Tenembaum) para lograr un Unix mínimo, capaz de correr un shell y un compilador por lo menos. Linus empezó escribiendo el núcleo del proyecto en ensamblador, y luego comenzó a añadir código en C, lo cual incrementó la velocidad de desarrollo, e hizo que empezara a tomarse en serio su idea de hacer un "MINIX mejor que MINIX". La versión 0.01 nunca llegó a ser compilada con éxito y la 0.02 ya presentaba mejoras. Luego Linus anunció en Internet su proyecto de la siguiente manera:

"Si suspiras al recordar aquellos días cuando lo hombres eran hombres y escribían sus propios controladores. Si te sientes sin ningún proyecto interesante y te gustaría tener un verdadero sistema operativo que pudieras modificar a placer, Si te resulta frustrante tener solo Minix. Entonces este artículo es para ustedes".
Y Linux fue liberado en Internet y la respuesta de los programadores y usuarios de UNIX fue contundente. Pronto todos querían aportar sus conocimientos para que la incipiente criatura fuera un sistema operativo estable, robusto y potente. Finalmente llegó la primera versión estable del Kernel la 1.0. De allí en adelante Linux fué evolucionando a un ritmo vertiginoso hasta hacer tambalear a todos los S.O del mercado. La versión actual del Kernel 2.0.33 estable y 2.1.97 experimental.

[9] www.linux.org

GNU[10] es un proyecto de software libre, que incluye el código fuente de los programas que entran dentro del proyecto y que desean llamarse "GNU", eso dió ideas a programadores de todo el mundo para cumplir sus deseos aunque Linus no los cumpliera y para portar (portar significa programar para otro tipo de plataforma o máquina) Linux a otra cosa como las computadoras Alpha de Digital. De ésta forma, todo aquél que quizo modificar al código pudo hacerlo, porque gracias a la licencia de uso del proyecto GNU eso se podia hacer sin problemas legales. De esta Linus se convirtió en el "moderador" de uno de los desarrollos más grandes en la historia del cómputo que ahora se llama Linux.

La versión 1.0 de Linux surgió en 1994 ofreciendo mejoras sobre el Unix tipico, incluyendo multitarea, memoria virtual, tcp/ip y alrededor de 175000 lineas de código. La versión 2.0 salió en junio de 1996 ofreciendo procesos de 64 bits, multiprocesamiento y tuvo 780000 lineas de código. La versión 2.1.110 julio de 1998 tuvo aproximadamente 1500 000 lineas de código 17% en especificación de arquitectura de código, 54% en controladores y 29% dedicado al kernel[11].

Linux es ahora más que un sistema operativo. El número de gente interesada en linux es cada vez mayor, la gente comparte información y código acerca de él. En agosto de 1999 se supo que 2500 personas habían contribuido en 3500 aplicaciones; procesadores de palabras, aplicaciones matemáticas y juegos.

Linus, además de seguir programando, se ha dedicado desde entonces a recopilar, aceptar, desechar, normar y organizar código de programación que le contribuyen otros programadores y a orientar y unir en equipos a grupos de programadores con ideas afines.

[10] www.gnu.org home.html
[11] Núcleo del sistema operativo

2.3 CARACTERÍSTICAS GENERALES DE LINUX.

Es un sistema operativo, el cuál tiene la función de administrar recursos de entrada, salida y periféricos así como tener control sobre la computadora. Se le llama kernel al módulo principal del sistema operativo, el cual es la primera parte que se carga en sistema operativo, manteniéndolo en la memoria principal del sistema.

Los elementos principales de Linux son:

- El núcleo del sistema operativo (núcleo o Kernel) , en términos estrictos éste es el sistema operativo, realiza tareas tales como el acceso a los dispositivos (terminales, discos, ratón etc.,).

- El intérprete de comandos (shell) es la interfaz básica que ofrece el sistema para su interacción con el usuario, puede engendrar procesos, que no es otra cosa que ejecutar programas, tiene su propio lenguaje de programación y muchas características que se mencionan más adelante, existen varios shells cada uno con sus particularidades, pero el que UNIX instala por defecto es el bourne shell.

- Utilerías varias; son programas de utilitarios básicos que normalmente acompañan al sistema operativo, los diferentes sabores de UNIX comparten la mayoría de las diferentes variantes que hay (incluyendo por supuesto a Linux), pero los cursos de administración cambian de marca en marca aunque comparten las bases.

De los puntos más destacables de éste sistema operativo están:

- Es multitarea: puede ejecutar varias tareas simultáneamente, es decir, no requiere que una termine para que otra pueda comenzar.

- Es multiusuario: Varios usuarios pueden trabajar de manera simultáneamente en el sistema.

- La Instalación está disponible en más de 40 idiomas distintos.

Las características nuevas de instalación incluyen la posibilidad de descargar e instalar las actualizaciones al momento de la instalación, un modo de "instalación mínima" para hacer caber a un sistema Linux en 65 MB de su disco duro.

Además, Linux soporta todo tipo de periféricos: CD-ROMS, grabadoras, tarjetas de sonido, tarjetas de red, etc..., utilizando todo tipo de soportes y buses (PCI, AGP, VESA), por lo que salvo que se tenga un periférico demasiado extraño (que no suele ser lo normal) puede ser configurado fácilmente para que funcione GNU/Linux. Tiene muchas características que le hacen ser un impresionante sistema operativo. De hecho, en la actualidad se está convirtiendo en un firme adversario de otros sistemas operativos más extendidos del momento (Windows 9X). La característica más importante de GNU/Linux es que es software libre; es decir, se distribuye con su código para que se pueda modificar.

Es por eso que hay varios "sabores" de Linux, como: Mandrake, Red Hat, Debian entre otros. El problema es que mucha gente se piensa que por ser gratis y estar hecho por gente en sus ratos libres va a ser un sistema operativo bastante ineficiente. Muchas veces se prefiere alguna distribución de Linux básicamente por el buen precio/rendimiento. Sin embargo, cuando compañías como IBM lo están utilizando día a día, es un buen indicador de que hay más razones que el dinero. Regularmente Linux es usado como:

Servidor

Ésta es la forma más usual de encontrar Linux. Casos típicos de servicios son: mail, webserver, dns, proxy, impresión, archivos, router y/o firewall. Lo interesante es que una sóla máquina Linux puede hacer todo esto y al mismo tiempo. Personalmente llegué a ver máquinas relativamente modestas que realizan una o más de estas funciones sin mayores problemas, lo que habla muy bien del producto como una excelente solución a un buen precio/rendimiento.

Estación de trabajo

Es raro encontrar una estación de tipo Unix fuera de alguna Universidad, donde el estándar son PC's x86 corriendo alguna versión de Windows o NT. Sin embargo, muchos de las estaciones Unix se están reemplazando por Linux. Linux cuenta con todo el software estándar que posee cualquier estación Unix comercial.

Super Computador

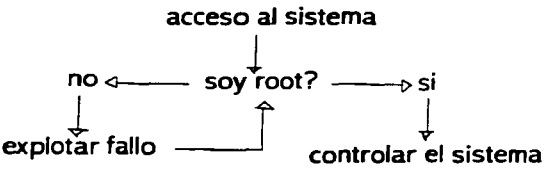
En el año 1994, se construyó en la NASA el primer super-computador basado en Linux. Éste consistio en 16 PC's 486, cada uno con 16MB RAM, 1GB en disco duro y dos tarjetas de red. Esto es lo que se conoce como un "cluster" y desde entonces, han surgido una gran cantidad de super computadores similares basados en Linux.

Las escenas de la famosa película "Titanic" fueron producidas en un cluster de máquinas Alpha corriendo Linux. Hoy en día las distribuciones de Linux incorporan estas características para armar clusters de servidores web, por ejemplo.

2.4 ENTRADA Y SALIDA DEL SISTEMA.

Ya que LINUX fue diseñado para trabajar con PC´s, desde el momento en que se conecta la computadora a la corriente eléctrica hasta que se obtiene el primer "login" pueden ocurrir muchas cosas y varias de ellas ocurren mucho antes de que el sistema operativo se de cuenta que se está ejecutando LINUX. Lo primero que se realiza es una autoprueba de encendido del hardware, esta función la hace el BIOS[12], que no es más que una ROM instalada en el computador, el control salta a una ubicación especifica y predefinida en RAM. Cuando se localiza el sector de arranque del disco duro, se cargan y ejecutan las instrucciones que ahí estén.

Para comenzar una sesión primeramente se debe tener acceso a la consola o a una terminal remota, esta última casi siempre en la que se tiene acceso y el uso de la consola se reserva al administrador. El sistema pide el identificador del usuario "login" para saber que usuario es el que solicita el acceso al sistema.



[12] Sistema básico de entrada y salida

Login:

El usuario escribe su login después de los dos puntos y presiona la tecla "enter", a lo que el sistema pide una contraseña para verificar la identidad del usuario, las contraseñas son privadas y sólo debe conocerlas el dueño, las contraseñas no son desplegadas cuando se escriben para proteger a los usuarios de personas ajenas.

Login: hector
Password:******
$
Si el identificador de usuario y contraseña han sido escritas correctamente el sistema mostrará un signo de pesos $ (este prompt puede cambiar de sistema en sistema o dependiendo de la configuración de las cuentas), se debe tener cuidado de cómo se teclean estas palabras, recuerde que mayúsculas y minúsculas son diferentes. Linux no es como DOS[13], a pesar de la similitud del indicador de comando, tienen poco en común.

Cuando el sistema está en la etapa de LOGIN, el siguiente paso que el sistema hace es el pedir una contraseña, y nos preguntaremos porque una contraseña; pues la contraseña es una forma de demostrar la identidad de un usuario al sistema, hay tres puntos para demostrar la identidad;

- Algo que se sabe.
- Algo que se tiene.
- Algo que se es

En computadoras personales la contraseña es una forma de identificar al usuario; los medios de seguridad siguen siendo físicos; tales como candados, puertas, paredes y sistemas biométricos, no hay contraseñas en sistemas

[13] Disk Operating System

monousuario. Hoy en día los sistemas UNIX cuentan con contraseñas para autenticar a los usuarios: el usuario tiene que usarla cada vez que quiera acceder al sistema. La contraseña no se despliega en texto claro. Generalmente, los sistemas UNIX usan el archivo passwd para registrar a cada usuario en el sistema. En el archivo passwd se tienen nombres de usuario, nombres reales, información de la cuenta de usuario e información de identificación. Uno de los blancos que persiguen los hackers aparte de ser root es el poder ver el archivo passwd, el cual se puede leer con el comando "cat".

En DOS o en cualquier otro sistema operativo monousuario, nosotros tenemos el control de todo, si decido apagar la máquina no pasa nada, sin embargo con varios usuarios trabajando en un sistema LINUX y docenas de recursos ocupados, no querrá apagar su computadora de manera repentina ya que desconcertaría a la gente conectada. Las primeras dos herramientas para apagar adecuadamente el sistema son vínculos hacia el mismo archivo: /sbin/halt y /sbin/reboot. La última de las herramientas es /sbin/shutdown.

La ejecución de "shutdown" con ciertos parámetros es en realidad la forma más segura de apagar el sistema. El comando shutdown sin atributos apaga el sistema pero no da un aviso a los usuarios, pero podemos configurar el "shutdown" a manera de que si el sistema va a recibir mantenimiento se les avise a los usuarios con media hora de anticipación, visto con una instrucción se vería de la siguiente manera:

shutdown -h +30 Sistema en mantenimiento

Este mensaje aparecería inmediatamente en la pantalla de todos y luego a intervalos cada vez más frecuentes hasta que el sistema finalmente se apagará. Para terminar la sesión, escriba exit, también puede funcionar logout o escribir control + D en el prompt.

TESIS CON
FALLA DE ORIGEN

2.5 AUTENTIFICACIÓN DE CONTRASEÑA EN LINUX

Autenticación clásica

En un sistema UNIX habitual cada usuario posee un nombre de entrada al sistema o login y una clave o password; ambos datos se almacenan generalmente en el fichero /etc/passwd. Tipicamente las versiones más nuevas de Linux implementan el uso de shadow, a veces puede obtenerse el /etc/shadow mediante xploits (programas que expotan vulnerabilidades), ya sea directamente o luego de obtener root.

Uno podría preguntarse qué utilidad tiene seguir más allá luego de tener root. Sucede que a veces una de las personas que es usuario en un determinado sistema es administrador o tiene alto privilegio en otro; por lo tanto, es importante conocer todos los passwords.

Este archivo contiene una línea por usuario (aunque hay entradas que no corresponden a usuarios reales, como veremos a continuación) donde se indica la información necesaria para que los usuarios se puedan conectar al sistema y trabajar en él, separando los diferentes campos:

toni:LEgPN8jqSCHCg:1000:100:Antonio Villalon,,,:/export/home/toni:/bin/sh

En primer lugar aparecen el login del usuario y su clave cifrada; a continuación tenemos dos números que serán el identificador de usuario y el de grupo respectivamente. El quinto campo, denominado gecos es simplemente información administrativa sobre la identidad real del usuario, como su nombre, teléfono o número de despacho.

Finalmente, los dos últimos campos corresponden al directorio del usuario (su $HOME inicial) y al shell que le ha sido asignado. Al contrario de lo que mucha gente cree, UNIX no es capaz de distinguir a sus usuarios por su nombre de

entrada al sistema. Para el sistema operativo lo que realmente distingue a una persona de otra (o al menos a un usuario de otro) es el UID del usuario en cuestión, el login es algo que se utiliza principalmente para comodidad de las personas (obviamente es más fácil acordarse de un nombre de entrada como toni que de un UID como 2643, sobre todo si se tienen cuentas en varias máquinas, cada una con un UID diferente). Por tanto, si en /etc/passwd existen dos entradas con un mismo UID, para UNIX se tratará del mismo usuario, aunque tengan un login y un password diferente: así, si dos usuarios tienen asignado el UID 0, ambos tendrán privilegios de superusuario, sin importar el login que utilicen.

Esto es especialmente aprovechado por atacantes que han conseguido privilegios de administrador en una máquina: pueden añadir una línea a /etc./passwd mezclada entre todas las demás, con un nombre de usuario normal pero con el UID 0; así garantizan su entrada al sistema como administradores en caso de ser descubiertos, por ejemplo para borrar huellas.

Como a simple vista puede resultar difícil localizar la línea insertada, especialmente en sistemas con un gran número de usuarios, para detectar las cuentas con privilegios en la máquina podemos utilizar la siguiente orden:

```
anita:~# awk -F: '$3==0 {print $1}' /etc/passwd
root
anita:~#
```

En el fichero de claves van a existir entradas que no corresponden a usuarios reales, sino que son utilizadas por ciertos programas o se trata de cuentas mantenidas por motivos de compatibilidad con otros sistemas; típicos ejemplos de este tipo de entradas son lp, uucp o postmaster. Estas cuentas han de estar bloqueadas en la mayoría de casos, para evitar que alguien pueda utilizarlas. Para acceder a nuestro sistema: sólo han de ser accesibles para el root

mediante la orden *SU*. Aunque en su mayoría cumplen esta condición, en algunos sistemas estas cuentas tienen claves por defecto o, peor, no tienen claves, lo que las convierte en una puerta completamente abierta a los intrusos, es conveniente que, una vez instalado el sistema operativo, y antes de poner a trabajar la máquina, comprobemos que están bloqueadas, o en su defecto que tienen claves no triviales. Algunos ejemplos de cuentas sobre los que hay que prestar una especial atención son: root, guest, lp, demos, 4DGifts, tour, uucp, nuucp, games o postmaster; es muy recomendable consultar los manuales de cada sistema concreto, y chequear periódicamente la existencia de cuentas sin clave o cuentas que deberán permanecer bloqueadas y no lo están.

Para cifrar las claves de acceso de sus usuarios, el sistema operativo UNIX emplea un criptosistema irreversible que utiliza la función estándar de C crypt(3), basada en el algoritmo DES[14], que recientemente fué cambiado por otro mecanismo mucho mejor denominado MD5 que es mucho más difícil de crackear (en realidad adivinar). Existen en la red crakeadores que funcionan con diccionarios de palabras, capaces de adivinar el password por fuerza bruta.

Para efectos de éste trabajo, nos limitaremos a mencionar que la función crypt toma como clave los ocho primeros caracteres de la contraseña elegida por el usuario (si la longitud de ésta es menor, se completa con ceros) para cifrar un bloque de texto en Claro de 64 bits puestos a cero; para evitar que dos pastores iguales resulten en un mismo texto cifrado, se realiza una permutación durante el proceso de cifrado elegida de forma automática y aleatoria para cada usuario, basada en un campo formado por un número de 12 bits (con lo que conseguimos 4096 permutaciones diferentes) llamado salt. El cifrado resultante se vuelve a cifrar utilizando la contraseña del usuario de nuevo como clave, y permutando con el mismo salt, repitiéndose el proceso 25 veces.

[14] Estandar de cifrado de datos

El bloque cifrado final, de 64 bits, se concatena con dos bits cero, obteniendo 66 bits que se hacen representables en 11 caracteres de 6 bits cada uno y que, junto con el salt, pasan a constituir el campo password del fichero de contraseñas, usualmente /etc./passwd. Así, los dos primeros caracteres de este campo estarán constituidos por el salt y los 11 restantes por la contraseña cifrada:

toni:LEgPN8jqSCHCg:1000:100:Antonio Villalon,,,:/export/home/toni:/bin/sh
salt: LE Password cifrado: gPN8jqSCHCg

Como hemos dicho antes, este criptosistema es irreversible. Entonces, ¿cómo puede un usuario conectarse a una máquina UNIX? El proceso es sencillo: el usuario introduce su contraseña, que se utiliza como clave para cifrar 64 bits a 0 basándose en el salt, leído en /etc/passwd, hemos preferido no mostrar las claves por defecto (si las tienen) ni el sistema operativo concreto.

Si tras aplicar el algoritmo de cifrado el resultado se corresponde con lo almacenado en los últimos 11 caracteres del campo password del fichero de contraseñas, la clave del usuario se considera válida y se permite el acceso. En caso contrario se le deniega y se almacena en un fichero el intento de conexión fallido.

Dentro de este apartado vamos a comentar brevemente la función de algunos servicios de Linux y sus potenciales problemas de seguridad. Los aquí expuestos son servicios que habitualmente han de estar cerrados; por lo que, no implican excesivos problemas de seguridad conocidos. Así, no vamos a entrar en muchos detalles con ellos; en puntos siguientes hablaremos con más extensión de otros servicios que suelen estar ofrecidos en todas las máquinas, como ftp, telnet o smtp, y que en su mayoría presentan mayores problemas de seguridad.

Autentificacion en el servicio FTP.

FTP (File Transfer Protocol, puerto 21 tcp) es, como su nombre indica, un protocolo de transferencia de ficheros entre sistemas. Desde un equipo cliente conectamos a un servidor para descargar ficheros desde él.

Un problema básico y grave de ftp es que está pensado para ofrecer la máxima velocidad en la conexión, pero ni mucho menos para ofrecer la máxima segundad; todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier fichero, se realiza en texto claro, con lo que un atacante lo tiene muy fácil para capturar todo ese tráfico y conseguir así un acceso válido al servidor. Incluso puede ser una amenaza a la privacidad de nuestros datos el hecho de que ese atacante también pueda capturar y reproducir los ficheros transféridos. Para solucionar este problema es conveniente concienciar a los usuarios de la utilidad de aplicaciones como scp y sftp, incluidas en el paquete ssh, que permiten transferir ficheros pero cifrando todo el tráfico; de esta forma, son el mejor sustituto de ftp.

Parece evidente que la conexión ftp a nuestro sistema ha de estar restringida a los usuarios que realmente lo necesiten: por ejemplo, un usuario como root en principio no va a necesitar utilizar este servicio, ya que por lo general va a trabajar en consola; otros usuarios considerados 'del sistema' (donde se incluye por ejemplo a postmaster, bin, uucp, shutdown, daemon.) tampoco necesitarán hacer uso de ftp.

Podemos indicar este tipo de usuarios a los que no les está permitida una conexión vía ftp a nuestra máquina en /etc/ftpusers, con un nombre por línea; un ejemplo de este fichero es el siguiente:

```
luisa:~# cat /etc/ftpusers
halt
operator
root
```

FTP anónimo

Los problemas relacionados con la seguridad del servicio ftp son especialmente preocupantes cuando se trata de configurar un servidor de ftp anónimo; las máquinas se pueden convertir en servidores de imágenes pornográficas o de warez (copias ilegales de programas comerciales). Conseguir un servidor de ftp anónimo seguro puede llegar a ser una tarea complicada: incluso en las páginas de ayuda de algunas variantes de Unix (como Solaris) se trata de facilitar el proceso para el administrador mediante un shellscript que (por defecto) presenta graves problemas de seguridad, ya que deja una copia del fichero de claves del sistema como un archivo de acceso público y anónimo.

Para configurar correctamente un servidor de este tipo necesitamos en primer lugar crear al usuario ftp en /etc/passwd y /etc/shadow, así como su directorio de conexión (algunos sistemas Unix, como Linux, ya incorporan esto al instalar el sistema). Este directorio ha de pertenecer a root (ningún fichero o subdirectorio ha de pertenecer nunca a ftp) y al grupo al que pertenece ftp: con esto conseguimos que los permisos de propietario sean para el administrador y los de grupo para los usuarios anónimos; estos permisos serán 555.

Dentro del $HOME de ftp hemos de crear el árbol de directorios mínimo para poder trabajar correctamente; esto es, debido a la llamada a chroot() que se utiliza en los accesos anónimos, que permite a esos usuarios ver el directorio raíz de su conexión en el directorio real ~ftp/. Al menos dos directorios son necesarios: etc/ y bin/, ambos propiedad de root y con modo 111. En el primero de ellos hemos de crear un fichero passwd y otro group, utilizados no con propósitos de autenticación sino para visualizar el propietario y grupo de cada fichero en el entorno sobre el que se ha aplicado chroot() al ejecutar ls: por tanto, no hace falta ninguna contraseña en ese fichero passwd, y sólo ha de contener entradas para los usuarios que posean ficheros bajo la jerarquía de ftp, como root; de la misma forma, el fichero group sólo ha de contener las entradas correspondientes a grupos que posean ficheros en dicha jerarquía:

```
anita:~# cat /export/home/ftp/etc/passwd
root:*:0:1:El Spiritu Santo:/:/sbin/sh
anita:~# cat /export/home/ftp/etc/group
root::0:
other::1:
daemon::2:
ftp::30000:
anita:~#
```

Autentificación en el servicio TELNET.

El protocolo telnet (tcp, puerto 23) permite utilizar una máquina como terminal virtual de otra a través de la red, de forma que se crea un canal virtual de comunicaciones similar (pero mucho más inseguro) al utilizar una terminal físicamente conectada a un servidor, la idea es sencilla: estamos accediendo remotamente en modo texto a un equipo igual que si estuviéramos utilizando su consola o una de sus terminales físicas, lo que nos permite aprovechar toda su potencia de cálculo sin necesidad de desplazamos hasta la ubicación de ese servidor, sino trabajando cómodamente desde nuestro propio equipo.

Telnet es el clásico servicio que hasta hace unos años no se solía deshabilitar nunca: no es habitual adquirir una potente máquina corriendo Unix y permitir que sólo se trabaje en ella desde su consola; lo más normal es que este servicio esté disponible para que los usuarios puedan trabajar remotamente, al menos desde un conjunto de máquinas determinado. Evidentemente, reducir al mínimo imprescindible el conjunto de sistemas desde donde es posible la conexión es una primera medida de seguridad; no obstante, no suele ser suficiente: recordemos que telnet no utiliza ningún tipo de cifrado, por lo que todo el tráfico entre equipos se realiza en texto claro. Cualquier atacante con un analizador de red (o un vulgar snifer) puede capturar el login y el password utilizados en una conexión; el snifing siempre es peligroso, pero más aún en sesiones telnet en las que transmitimos nombres de usuarios y contraseñas:

estamos otorgando a cualquiera que lea esos datos un acceso total a la máquina destino, bajo nuestra identidad. Por tanto, es muy recomendable no utilizar telnet para conexiones remotas, sino sustituirlo por aplicaciones equivalentes pero que utilicen cifrado para la transmisión de datos: ssh o SSL-Telnet son las más comunes. En estos casos necesitamos además de la parte cliente en nuestro equipo, la parte servidora en la máquina remota escuchando en un puerto determinado.

Aparte del problema de los atacantes husmeando claves, los demonios telnetd han sido también una fuente clásica de problemas de programación, cualquier versión de este demonio que no esté actualizada es una potencial fuente de problemas, por lo que conviene conseguir la última versión de telnetd para nuestro Unix particular, especialmente si aún tenemos una versión anterior a 1997.

Otros problemas, como la posibilidad de que un atacante consiga recuperar una sesión que no ha sido cerrada correctamente, el uso de telnet para determinar qué puertos de un host están abiertos, o la utilización del servicio telnet (junto a otros, como ftp) para averiguar el clon de Unix concreto (versión de kernel incluida) que un servidor utiliza, también han hecho famosa la inseguridad de este servicio. Antes hemos hablado de la configuración de un entorno restringido para usuarios ftp invitados, que accedan mediante su login y su contraseña pero que no veían la totalidad del sistema de ficheros de nuestra máquina. Es posible hacer algo parecido con ciertos usuarios interactivos, usuarios que se conectarán al sistema mediante telnet utilizando también su login y su password, pero que no verán el sistema de ficheros completo: sólo la parte que a nosotros nos interese (en principio).

Para que un usuario acceda mediante telnet a un entorno restringido con chroot() necesitamos en primer lugar un entorno parecido al que hemos visto antes: a partir de su directorio $HOME, una serie de subdirectorios (bin/, lib/,

etc/. . .) dentro de este último existiría al menos un fichero group y otro passwd, no se usan con propósitos de autenticación; por lo que, no es necesario que existan claves reales en ninguno de ellos.

2.6 COMANDOS EN LINUX

En Linux se invoca un comando escribiendo su nombre y separados por blancos, los argumentos opcionales. Como antes, se debe pulsar "enter" una vez se ha escrito la orden correspondiente. Se mencionarán algunos de los comandos que nos servirán para analizar y detectar intrusos, comenzando con los básicos:

El comando "who" informa de los usuarios que se hallan presentes en el sistema, éste comando es básico en el mundo de los hackers y también un "arma de doble filo" ya que "who" puede ser usado como troyano, así como también nos sirve para detectar intrusos.

"who"

Jose ttyp Ene 27 10:45
Marcelo ttyp Ene 27 11:34
Hector ttyp Feb 17 9:43

Si deseamos saber quienes somos (cara al sistema). "who am i"

"cat"

Concatena archivos. A veces, también se utiliza para mostrar un archivo. Algunas de las opciones para el comando cat tienen especificadores de opción tanto largos como cortos.

"cd"
Cambia el directorio de trabajo actual.

Sintaxis:
Cd nombre del directorio

"chgrp"
Cambia la propiedad de grupo de un archivo. Algunas de las opciones para el comando chgrp tienen especificadores de opción tanto largos como cortos.

Sintaxis:
Chgrp (opciones) grupo listaarchivo
Chgrp ventas /usr/cosadeventas/*

Este ejemplo cambia todos los archivos que se encuentran en el directorio "cosasdeventas" a la propiedad del grupo ventas.

"Chmod"

Chmod cambia el modo de los archivos. El modo de un archivo controla los permisos de
acceso asociados con ese archivo. Linux tiene tres niveles de seguridad el propietario, acceso al grupo y los demás. Dentro de estos tres niveles hay tres permisos: lectura, escritura y ejecución.

ls
Descripción: =list. listar contenido de directorios.
Ejemplos: ls, ls -l, ls -fl, ls —color

Los comandos son muy útiles, pero con el conocimiento básico del shell y sus comandos tenemos armas muy poderosas que muestran todo el potencial del interprete de comandos Unix. A continuación se muestran algunos ejemplos de comandos que nos sirven para la administración del sistema, así como también para la detección de acciones maliciosas.

history
Descripción: muestra el historial de comandos introducidos por el usuario.
Ejemplos: history | more

more
Descripción: muestra el contenido de un fichero con pausas cada 25 lineas.
Ejemplos: more fichero

lynx
Descripción: navegador web con opciones de ftp, https.
Ejemplos: lynx www.cert.org, lynx –source
http://www.cert.org/script.sh | sh

head
Descripción: muestra la cabecera (10 lineas) de un fichero.
Ejemplos: head fichero, head -100 /var/log/maillog | more

ping
Descripción: heramienta de red para comprobar entre otras cosas si llegamos a un host remoto.
Ejemplos: ping tigre.aragon.unam.mx

uname
Descripción: =unix name. Información sobre el tipo de unix en el que estamos, kernel, etc.
Ejemplos: uname, uname -a

ulimit
Descripción: muestra los limites del sistema (máximo de ficheros abiertos, etc..)
Ejemplos: ulimit

adduser
Descripción: añadir usuario de sistema.
Ejemplos: adduser pepe, adduser -s /bin/false pepe

usermod
Descripción: = modificar usuario de sistema
Ejemplos: usermod -s /bin/bash pepe

df
Descripción: = disk free. espacio en disco disponible. Muy útil.
Ejemplos: df, df -h

netstat
Descripción: la información sobre las conexiones de red activas.
Ejemplos: netstat, netstat -ln, netstat -l, netstat -a

traceroute
Descripción: herramienta de red que nos muestra el camino que se necesita para llegar a otra maquina.
Ejemplos: traceroute www.unam.mx

ifconfig
Descripción: =interface config. configuración de interfaces de red, modems, etc.
Ejemplos: ifconfig, ifconfig eth0 ip netmask 255.255.255.0

route
Descripción: gestiona las rutas a otras redes.
Ejemplos: route, route -n

Descripción: Sniffer o husmeador de todo el tráfico de red. No suele venir instalado por defecto.
Ejemplos: sniffit -i

2.7 ARCHIVOS Y DIRECTORIOS.

Uno de los primeros manuales de Unix, establecía que en Unix todos son archivos, tanto ficheros normales, como directorios;

Fichero normal: con sus variantes que más tarde veremos. Es algo que, referido a un nombre, contiene una secuencia determinada de caracteres. El sistema operativo no impone ningún tipo de formato ni de registro. Ejemplos de archivos normales pude ser el que no creamos con algún editor, conteniendo un documento. Por ejemplo el fichero /etc/motd es un fichero cuyo contenido es el mensaje del día.

Directorios: Son archivos cuyo contenido son nombres de arribos funcionalmente, se comportan de la misma manera que en sistemas tipo MS-DOS.

Tipos de archivos

Si bien UNIX no impone una estructura a ningún fichero, estos tendrán características comunes dependiendo para lo que sirvan: podemos agrupar estos en varios tipos:

Ejecutables:

Normalmente se trata de programas compilados y contienen código binario ininteligible para la mayoría de los humanos pero no así para la máquina. Ejemplos de estos archivos pueden ser los /bin/ls , /bin/cat/ todos ellos deben tener activados los permisos de ejecución que mas tarde veremos.

Binarios:

Englobando dentro de esta categoría aquellos que son empleados por programas capaces de entender su contenido, pero no legibles.

Texto:

Correspondientes a aquellos archivos que contienen registros de caracteres terminados en nueva línea y normalmente son legibles. Ejemplo de fichero de texto puede ser el /etc/motd /etc/passwd y cualquiera que haya sido confeccionado con el editor.

Shells:

Ésta es otra característica que hace de UNIX y en consecuencia de Linux, lo que es: el sistema operativo más flexible, aunque en los últimos años se han agregado interfaces gráficas al sistema UNIX, casi todas las utilerías para emplear y administrar Linux, se ejecutan mediante la escritura de comandos. En Linux, al intérprete de la línea de comandos de le conoce como shell, que no es otra cosa más que un programa diseñado para aceptar comandos y ejecutarlos. Varios tipos de programas pueden emplearse como shells, pero en casi todas las versiones de Linux existen diversos shells estándares disponibles.

Los shells de Linux, son equivalentes al COMMAND.COM que emplea Windows. Ambos aceptan y ejecutan comandos, y corren archivos de procesamiento por lotes y programas.

Dispositivo:
Cualquier fichero asignado a un dispositivo físico: normalmente residen a partir del subdirectorio /dev y son archivos de terminales.

En cualquier sistema multiusuario es preciso que existan métodos que impidan que un determinado usuario pueda modificar o borrar un fichero confidencial, o incluso leer su contenido. Asimismo, determinados comandos deben estar permitidos exclusivamente a determinados usuarios, quedando inoperantes para los demás.

En UNIX estos métodos radican en que cada fichero tiene un propietario, que es el usuario que creó el fichero. Además los usuarios están divididos en grupos, asignación que normalmente se lleva a cabo por el administrador de sistemas, dependiendo de la afinidad de las tareas que se realizan.

Un fichero puede tener cualquier combinación de los tres tipos de acceso sobre tres tipos de usuarios: el creador, los de su grupo y todos los demás, otros cualquiera que no cumplan ninguna de las dos condiciones anteriores. Para ver los permisos de un fichero cualquiera, empleamos el comando ls -l (formato largo)

```
Sls -l
-rw-r-1 jose sys 4 mar
drw-r-r-1 jose sys
```

2.8 EL LIBRO NARANJA.

Se pueden entender como seguridad una característica de cualquier sistema de cómputo que indica que está libre de la mayoría de peligros, daños o riesgos, y que es, en cierta manera confiable. Como ya se mencionó, el mantener un sistema seguro consiste básicamente en garantizar la confidencialidad, integridad y la disponibilidad . Aunque cada entidad suele dar mayor prioridad a alguno de estos aspectos, es impredecible, que estén presentes los tres, con frecuencia se habla del nivel de seguridad de un sistema o red pero no siempre se hace referencia a estándares mundiales. El departamento de defensa de los Estados Unidos definió en 1983 niveles de seguridad para sus computadoras; los cuales están registrados en el denominado "orange book[15]" y es usado como un estándar para indicar el nivel de seguridad de los sistemas informáticos, el cual establece criterios para medir la fiabilidad de los sistemas informáticos en seguridad. Antes de mencionar los diferentes niveles de seguridad, es necesario conocer algunos conceptos relevantes:

- Base fiable de cómputo (TCB): Es el conjunto de mecanismos relevantes a efectos de la seguridad del sistema. En los niveles con una fiabilidad elevada, la TCB se construye en torno a un monitor de referencias que impone las relaciones de acceso autorizadas entre los sujetos y objetos de un sistema.

- Control de accesos discrecional: permite restringir el acceso a los objetos basándose en la identidad de los usuarios y/o grupos de usuarios a los que pertenecen. Los usuarios protegen sus objetos indicando quien puede acceder y el tipo de acceso permitido.

- Reutilización de objetos. implica proteger archivos, memoria y otros objetos de accesos por parte de un usuario tras su uso por otro. Por ejemplo un usuario crea un archivo en el que almacena información confidencial y después la borra, a continuación otro usuario crea un archivo en el que almacena información

[15] Tambien llamado libro naranja. www.dynamoo.com orange

confidencial y después borra. Después otro usuario malicioso reserva espacio en el disco y el sistema le asigna esos mismos bloques, si el sistema no borra físicamente la información del usuario anterior, el otro usuario podría leer la información borrada por el dueño original.

- Etiquetas: las etiquetas de confidencialidad se asocian a cada sujeto y a cada objeto (archivo, directorio) e indican su nivel de autoridad asociado y se denomina habilitación. La etiqueta de confidencialidad de un archivo especifica el nivel de autoridad que un usuario debe tener para acceder al mismo.

- Identificación y autentificación: es necesario que los usuarios se identifiquen antes de realizar cualquier actividad que implique una interacción con la TCB (ejecutar un programa, leer un archivo).

- Distribución segura: garantiza la protección del sistema mientras se envía a un cliente, asegurando que el sistema que recibe es idéntico al suministrado por el vendedor.

- Arquitectura del sistema: están relacionados con el diseño de un sistema para que sea posible la seguridad.

- Vía fiable: en algunos sistemas se requiere que los usuarios puedan conectarse desde una terminal al sistema a través de lo que se llama una vía fiable. Para ello, existe una secuencia de teclas, que al pulsarse elimina todos los procesos actuales y establece una conexión segura con la TCB permitiendo su autentificación. Esto evita ataques sistemáticos contra el sistema mediante programas marcadores y la introducción de caballos de troya en los programas de conexión al sistema.

Niveles de seguridad

El libro naranja divide su clasificación en cuatro niveles de seguridad. Los requisitos para un determinado nivel siempre lo son para el siguiente, pudiendo éste restringir más aun los criterios, ya que se trata de una jerarquía de niveles:

Nivel D (seguridad mínima). En esta categoría están englobados todos los sistemas que han sido valorados y no han superado los requisitos mínimos para pertenecer a un nivel de seguridad superior. En esta categoría no existen requisitos de seguridad.

Computadoras bajo MS-DOS o las versiones personales de windows 9x, además de otros sistemas antiguos son un ejemplo de sistema que pertenecen a esta categoría.

NIVEL C1 (protección mediante seguridad discrecional). Todos los usuarios manejan los datos al mismo nivel. En este nivel se procura evitar que los usuarios cometan errores y dañen al sistema. Las características más importantes de este nivel son el control de autentificación mediante contraseñas y la protección discrecional de los objetos. el código del sistema debe estar protegido frente a ataques procedentes de programas de usuario.

Un sistema de este nivel no necesita distinguir entre usuarios individuales, tan sólo entre tipos de accesos permitidos o rechazados. En este nivel hay que ser dueño de un objeto para ceder sus derechos de accesos y siempre se protege a los objetos de nueva creación.

Nivel C2 (protección mediante accesos controlados).

A partir de este nivel, el sistema debe ser capaz de distinguir entre usuarios individuales. Generalmente el usuario debe ser dueño de un objeto para ceder los derechos de acceso sobre él. En la mayoría de los sistemas Unix a partir de este nivel, existen listas de control de acceso (acls). Debe permitir que los

recursos del sistema se protejan mediante accesos controlados. En Unix el acceso a los periféricos (dispositivos de e/s) sigue un esquema de permisos idéntico al de los archivos de los usuarios.

Se aplican los requisitos de reutilización de objetos cuando ésos mismos se reasignan. Se requiere a partir de éste nivel que el sistema disponga de auditoría; por ello, cada usuario debe tener un identificador único que se utiliza para comprobar todas las acciones solicitadas, se deben auditar todos los sucesos relacionados con la seguridad y proteger la información de la auditoría. El sistema debe ser capaz de auditar a nivel de usuario, la mayor parte de los Unix comerciales pertenecen a este nivel, puesto que lo único que han tenido que añadir los fabricantes es un paquete de auditoria.

NIVEL B1 (protección mediante seguridad etiquetada)
A partir de este nivel, los sistemas poseen un control de accesos obligatorio que implica colocar una etiqueta a los objetos (principalmente sobre los archivos). Esto, junto con el nivel de habilitación de los usuarios es utilizado para reforzar la política de seguridad del sistema. En estos sistemas, el dueño no es el responsable de la protección del objeto, a menos que disponga de la habilitación necesaria. En cuánto a la auditoria, el sistema debe ser capaz de registrar cualquier cambio o anulación en los niveles de seguridad, y también hacerlo selectivamente por nivel de seguridad

Debe existir una documentación que incl[illegible] l modelo de seguridad soportado por el sistema.

No es necesaria una demostración matemática, pero si una exposición de las reglas implantadas por las características de seguridad del sistema.

Nivel B2 (protección estructurada). A partir de este nivel los cambios en los requisitos no son visibles desde el punto de vista del usuario respecto a los niveles anteriores.

En B2, todos los objetos del sistema están etiquetados, incluidos los dispositivos. Deben existir vías fiables que garanticen la comunicación segura entre un usuario y el sistema. Los sistemas deben ser modulares y utilizar componentes físicos para aislar las funciones relacionadas con la seguridad de las demás. Requieren una declaración formal del modelo de seguridad del sistema, y que haya una gestión de la configuración. También deben buscarse los canales ocultos.

Nivel B3 (dominios de seguridad). Es necesario que exista un administrador de seguridad, que sea alertado cuando se detecta una violación inminente de la seguridad. Deben existir procedimientos para garantizar que la seguridad se mantiene aunque el servidor se caiga y luego reinicie. Es obligatoria la existencia de un monitor de referencia sencillo, a prueba de agresiones e imposible de eludir. La TCB debe excluir todo el código fuente que no sea necesario para proteger el sistema.

NIVEL A1 (diseño verificado). Esta clase de certificación más alta, aunque el libro naranja no descarta la posibilidad de exigir requisitos adicionales. Son sistemas funcionalmente equivalentes a B3. Sólo se añade la distribución fiable que refuerza la seguridad. Los sistemas A1 tienen la confiabilidad adicional que ofrece el análisis formal y la demostración matemática de que el diseño del sistema cumple el modelo de seguridad y sus especificaciones de diseño.

2.9 VULNERABILIDADES EN LINUX.

La comunidad de usuarios Linux se sienten orgullosos de la estabilidad e invulnerabilidad de linux, ante otros sistemas operativos que son presas de ataques con virus.

Por otro lado, las herramientas para atacar servidores Linux existen por toda la Internet, y existe más de una para atacar un mismo servicio. Por ejemplo: se ha descubierto que utilizando *scripts* de Flash en páginas web es posible teclear comandos de manera remota. Sólo faltaba que un programador creara una herramienta automática que explotara estas debilidades y que tuviera las mismas características de auto-replicación de un virus, de ahí viene la creación del gusano Ramen.

Ramen es un conjuto de herramientas que explotan vulnerabilidades bien conocidas en tres paquetes de software comunes. Este gusano ataca principalmente al Red Hat Linux en sus versiones 6.2 y 7.0, porque en estas versiones es que vienen activos por defecto los servicios que el gusano explota para romper la seguridad del servidor.

Los paquetes de software vulnerables son:

- wu-ftpd - Un error en la validación de una cadena en la función site_exec()
- rpc.statd - Error de sobreescritura de la pila de formato de cadena
- lprng - Puede pasar entradas del usuario en cadenas como parámetros a las llamadas de syslog().

Es posible detectar cuando el gusano intenta infectar un sistema si se tiene una herramienta para detectar revisiones de puerto, tales como *PortSentry* o *Snort*. worm husmea desde un sistema infectado los puertos TCP 21 (wu-ftpd – FTP).

El virus explota cualquiera de estos servicios y obtiene privilegios root. Una vez comprometido el sistema, se copia a sí mismo en el directorio */usr/src/.poop*, donde inicia una serie de actividades en el servidor local e inicia la búsqueda de nuevos servidores que infectar. La ventaja de Linux es que restaurar un servidor muchas veces no significa más que copiar nuevamente los binarios limpios y archivos de configuración, y finalmente reiniciar los servicios.

Sólo habrá que rastrear y eliminar las copias del Ramen que estén presentes en el servidor. Para evitar ser infectado habrá que eliminar el servidor ftp anónimo junto con cualquier otro servicio que no sea utilizado, conseguir las últimas versiones de los servicios afectados, y configurar los archivos de acceso de tal forma que no permita conexiones de servidores no confiables.
Dicho gusano nos hace ver que no hay que confiarse de la fama de seguro de un sistema. Cada día se descubren debilidades a sistemas viejos y nuevos.

Existen varias vulnerabilidades en diferentes versiones de linux, se mencionarán algunas en SuSe y también aplicaciones que son vulnerables.

Paquete: pine

Fecha: Enero 2003
Versiones afectadas: 7.2, 7.3, 8.0, 8.1, 8.2
SuSE eMail Server III, 3.1
SuSE Linux Enterprise Server 7, 8
SuSE Linux Firewall on CD/Admin host
SuSE Firewall on CD 2
SuSE Linux Connectivity Server
SuSE Linux Office Server
SuSE Linux Desktop 1.0

Pine es uno de los paquetes de linux que frecuentemente recibe ataques de intrusos, un hacker puede conectarse remotamente para escribir correo, pero realmente esta ejecutando comandos.

Otras vulnerabilidaes en las distribuciones de SuSe son:

- gdm2
The Gnome Display Manager (GDM) contiene un bug que permite a los atacantes ver cualquier archivo del sistema.

- whois

Existe un overflow en el commando "whois".

- xfs

El servidor (xfs) contiene varios overflows, que podrían permitir a usuarios remotos ejecutar commandos de administrador. Aunque SuSE no activa por default (xfs) es conveniente actualizar los paquetes.

- postgresql

SQL servidor de bases de datos también es vulnerable en la version 7.3.

Paquete: sendmail

Date: Abril 2003

Versiones afectadas: 8.0, 8.1, 8.2

SuSE Linux Enterprise Server 8

- traceroute
Hay un overflow en traceroute, puede ser fácilmente localizado por los atacantes para ganar acceso a un socket.

- gdm2
Es possible que los usuarios locales puedan leer cualquier texto o archivo creando un symlink desde ~/.xsession-errors.

Por lo general, las vulnerabilidades que aprovechan los hackers están en las aplicaciones que se hacen para el sistema operativo, también existen los exploits (programas explotadores de estas vulnerabilidades) actualmente se acaba de certificar Linux SuSe, como el sistema operativo más seguro, capaz de transferir grandes cantidades de información financiera de manera sumamente segura. Con ésta panorámica del funcionamiento del sistema operativo Linux, pasaremos a conocer uno de los mecanismos de detección de intrusos.

Capítulo III
SISTEMAS DE DETECCIÓN DE INTRUSOS.

Ahora que sabemos que existen "agujeros" en los sistemas operativos y en especial en Linux, es causa de preocupación; que los intrusos aprovechan dichos, para acceder a cualquier computadora. Las preguntas obligadas que deberíamos preguntarnos serían; ¿cómo puedo proteger mi sistema de ello?, ¿existen herramientas para tener seguro mi sistema?.

El Pentágono, la CIA[16], la ONU[17] y demás organismos mundiales han sido víctimas de intromisiones por parte de estas personas que tienen muchos conocimientos en la materia y también una gran capacidad para resolver los obstáculos que se les presentan, y ello sólo es un ejemplo de las víctimas que se mencionan; es por ello, que nace la necesidad de crear herramientas y métodos para la detección de los accesos no autorizados.

3.1 FLUJO DE LA INFORMACIÓN.

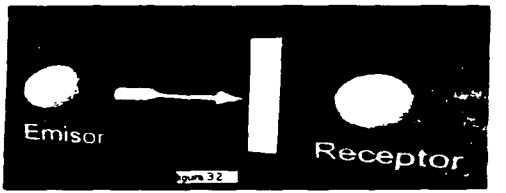
La detección de intrusos es el monitoreo de eventos que ocurren en un

Figura 3.1

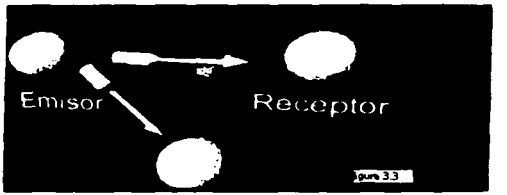
[16] Central Intelligence Agency
[17] Organización de las Naciones Unidas

sistema de cómputo, la detección de intrusos es relativamente joven pero realmente existe desde 1980. En 1999 la gente pensaba que UNIX era un sistema operativo relativamente seguro, estudios han permitido el aumento de la fiabilidad del sistema operativo, ya que muchas de las fallas de seguridad de UNIX se resuelven al tiempo que se hacen públicas. El súper usuario o administrador UNIX sigue siendo el blanco escencial de los hackers; cualquier intruso puede convertirse en superusuario y controlar así el sistema. Considerando el flujo de información de un emisor a un receptor, como se muestra en la figura 3.1, se puede obtener la clasificación de los diferentes tipos de ataques a un sistema, de la siguiente manera;

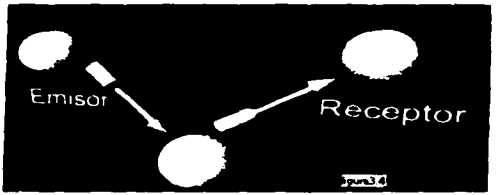
Interrupción. Se presenta cuando un recurso del sistema es destruido o se vuelve no disponible. Este es un ataque contra la disponibilidad. Ejemplos de este ataque son la destrucción de un elemento de hardware, como un disco [illegible]ar una línea de comunicación o desactivar el sistema administrador de archivos.



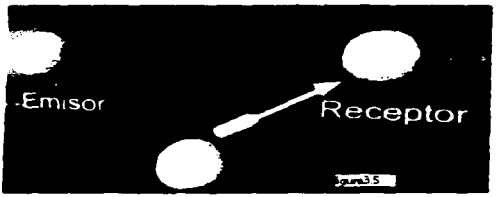
Interceptación. Se presenta cuando un agente no autorizado, ya sea una persona, programa o computadora, consigue acceder a un recurso (fig. 3.3). Ejemplos de este ataque son interceptar una línea para obtener datos que circulen por la red, la lectura de las cabeceras de paquetes para determinar la identidad de uno o más de los usuarios implicados en la comunicación (interceptación de identidad), o bien la copia ilícita de archivos o programas (interceptación de archivos o programas).



Modificación. Se presenta cuando un agente no autorizado, no sólo consigue acceso a un recurso, sino que es capaz de modificarlo. Este es un ataque contra la integridad. Ejemplos de este ataque son el cambio de valores en un archivo de datos, alterar el funcionamiento de un programa o modificar el contenido de mensajes que están siendo transferidos por la red. Como se muestra en la siguiente figura.



Fabricación. Se presenta cuando un agente no autorizado inserta objetos falsificados en el sistema. Este es un ataque contra la autenticidad. Ejemplos de este ataque son la inserción de mensajes falsos en una red, el añadir registros a un archivo, ver figura 3.5.



3.2 IMPORTANCIA DE LOS SISTEMAS DE DETECCIÓN DE INTRUSOS (IDSes)

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

Una de las primeras cosas que deberíamos plantearnos a la hora de hablar de IDSes es si realmente necesitamos de uno para nuestro entorno de trabajo; a fin de cuentas, debemos tener ya un sistema de protección basado en firewall, y por si nuestro firewall fallara, cada sistema habrá de estar configurado de una manera correcta, de forma que incluso sin firewall cualquier máquina pudiera seguirse considerando relativamente segura. La respuesta es, sin duda, *sí*; debemos esperar que en cualquier momento alguien consiga romper la seguridad de nuestro entorno informático, y por tanto hemos de ser capaces de detectar ese problema tan pronto como sea posible (incluso antes de que se produzca, cuando el potencial atacante se limite a probar suerte contra nuestras máquinas). Ningún sistema informático puede considerarse completamente seguro, pero incluso aunque nadie consiga violar nuestras políticas de seguridad, los sistemas de detección de intrusos se encargarán de

mostrarnos todos los intentos de multitud de piratas para penetrar en nuestro entorno, no dejándonos caer en ninguna falsa sensación de seguridad: si somos conscientes de que a diario hay gente que trata de romper nuestros sistemas, no caeremos en la tentación de pensar que nuestras máquinas estarán seguras porque nadie sabe de su existencia o porque no son interesantes para un hacker.

AROUITECTURA DE UN IDSes.

A grandes rasgos la arquitectura de un sistema de detección de intrusos es la siguiente:

- La fuente de recolección de datos.

- Reglas que contienen los datos y patrones para detectar anomalías de seguridad, cabe mencionar que las reglas van a depender del tipo IDS que se tenga, como ya se ha mencionado que hay distintos.

- Filtros que comparan los datos capturados de la red o de los logs con los patrones almacenados en el conjunto de reglas.

- Dispositivo generador de informes y alarmas. En algunos casos con la sofistificaciòn suficiente como para enviar alertas via mail, estaríamos hablando de detección automática de intrusos. Un IDS capaz de tomar acciones al momento de encontrar un intruso; es llamado sistema automático de detección de intrusos inteligente.

EVOLUCIÓN DE LOS IDSes

Desde los años 70's se empezaron a desarrollar algunos sistemas para detectar intrusos, podemos comencionar a Denning Neumann con el Intrusion Detection Expert System (IDES) en 1984, el cual era un modelo de detección

en tiempo real, que se encargaba de encontrar cosas raras o inusuales en el sistema por medio de técnicas estadísticas. El IDES fue usado en un sistema híbrido. Sytec fue otro proyecto nacido alrededor de 1985 utilizando bases de datos para la detección.

Después surgió Haystack que fue desarrollado por tracor Applied Sciences en 1989 y laboratorios haystack. Fue asignado para ayudar a la fuerza aérea en la seguridad; no obstante, este es uno de los campos con más auge, desde hace ya unos años dentro de la seguridad informática. Y no es extraño: la capacidad para detectar y responder ante los intentos de ataque contra nuestros sistemas es realmente muy interesante. Durante estos veinte años, cientos de investigadores de todo el mundo han desarrollado, con mayor o menor éxito, sistemas de detección de todo tipo, desde simples procesadores de bitácoras hasta complejos sistemas distribuidos, especialmente vigentes con el auge de las redes de computadores en los últimos años.

3.3 CLASIFICACIÓN DE LOS IDSes

Generalmente existen dos grandes enfoques a la hora de clasificar a los sistemas de detección de intrusos: o bien en función de que sistemas vigilan, o bien en función de cómo lo hacen. Si elegimos la primera de estas aproximaciones tenemos dos grupos de sistemas de detección de intrusos: los que analizan actividades de una única máquina en busca de posibles ataques, y los que lo hacen de una subred (generalmente, de un mismo dominio) aunque se emplacen en uno solo de los hosts de la misma. Esta última puntualización es importante: un IDS que detecta actividades sospechosas en una red no tiene porqué (y de hecho en la mayor parte de casos no suele ser así) ubicarse en todas las máquinas de esa red.

Vamos a hablar un poco acerca de los IDSes basados en red:
Un IDS basado en red, monitorea los paquetes que circulan por nuestra red en busca de elementos que denoten un ataque contra alguno de los sistemas

ubicados en ella; el IDS puede situarse en cualquiera de los hosts o en un elemento que analice todo el tráfico (como un HUB o un enrutador).

Esté donde esté, monitorear a diversas máquinas y no una sola: esta es la principal diferencia con los sistemas de detección de intrusos basados en host. IDSes basados en máquina. Mientras que los sistemas de detección de intrusos basados en red operan bajo todo un dominio de colisión, los basados en máquina realizan su función protegiendo un único sistema; de una forma similar guardando las distancias, por supuesto a como actúa un escudo antivirus residente en MS-DOS, el IDS es un proceso que trabaja en segundo plano (o que despierta periódicamente) buscando patrones que puedan denotar un intento de intrusión y alertando o tomando las medidas oportunas en caso de que uno de estos intentos sea detectado.

Algunos autores dividen el segundo grupo, el de los sistemas de detección de intrusos basados en máquina, en tres subcategorías:

Verificadores de integridad del sistema (SIV).
Un verificador de integridad no es más que un mecanismo encargado de monitorear archivos de una máquina en busca de posibles modificaciones no autorizadas, por norma general puertas traseras dejadas por un intruso (por ejemplo, una entrada adicional en el fichero de contraseñas o un /bin/login que permite el acceso ante cierto nombre de usuario no registrado). El SIV más conocido es sin duda Tripwire, comentado en este mismo trabajo; la importancia de estos mecanismos es tal que en la actualidad algunos sistemas Unix incluyen verificadores de integridad, como Solaris y su ASET (Automated Security Enhancement Tools).

Monitores de registros (LFM).

Estos sistemas monitorizan los archivos de bitácora generados por los programas (generalmente demonios de red) de una máquina en busca de patrones que puedan indicar un ataque o una intrusión. Un ejemplo de monitor puede ser swatch, pero más habituales que él son los pequeños shellscripts que casi todos los administradores realizan para comprobar periódicamente sus archivos de bitácora en busca de entradas sospechosas (por ejemplo, conexiones rechazadas en varios puertos provenientes de un determinado host, intentos de entrada remota como root.).

Sistemas de decepción.

Los sistemas de decepción o tarros de miel (honeypots), como Deception Toolkit (DTK), son mecanismos encargados de simular servicios con problemas de seguridad de forma que un pirata piense que realmente el problema se puede aprovechar para acceder a un sistema, cuando realmente se está aprovechando para registrar todas sus actividades. Se trata de un mecanismo útil en muchas ocasiones (por ejemplo, para conseguir 'entretener' al atacante mientras se rastrea su conexión) pero que puede resultar peligroso:

¿Qué sucede si el propio sistema de decepción tiene un bug que desconocemos, y el atacante lo aprovecha para acceder realmente a nuestra máquina?.

Realmente esta división queda algo pobre, ya que cada día se avanza más en la construcción de sistemas de detección de intrusos basados en host que no podrían englobarse en ninguna de las subcategorías anteriores. Otra gran clasificación de los IDSes se realiza en función de cómo actúan estos sistemas; actualmente existen dos grandes técnicas de detección de intrusos: las basadas en la detección de anomalías (anomaly detection) y las basadas en la detección de usos indebidos del sistema (misuse detection).

Aunque más tarde se hablará con mayor profundidad de cada uno de estos modelos, la idea básica de los mismos es la siguiente:

Detección de anomalías.

La base del funcionamiento de estos sistemas es suponer que una intrusión se puede ver como una anomalía de nuestro sistema, por lo que si fuéramos capaces de establecer un perfil del comportamiento habitual de los sistemas seríamos capaces de detectar las intrusiones por pura estadística: probablemente una intrusión sería una desviación excesiva de la media de nuestro perfil de comportamiento.

Detección de usos indebidos.

El funcionamiento de los IDSes basados en la detección de usos indebidos presupone que podemos establecer patrones para los diferentes ataques conocidos y algunas de sus variaciones; mientras que la detección de anomalías conoce lo normal (en ocasiones se dice que tienen un "conocimiento positivo", (positive knowledge) y detecta lo que no lo es, este esquema se limita conocer lo anormal para poderlo detectar (conocimiento negativo, negative knowledge).

Para ver más claramente la diferencia entre ambos esquemas, imaginemos un sistema de detección basado en monitoreo de las máquinas origen desde las que un usuario sospechoso se conecta a nuestro sistema: si se tratara de un modelo basado en la detección de anomalías, seguramente mantendrá una lista de las dos o tres direcciones más utilizadas por el usuario legítimo, alertando al responsable de seguridad en caso de que el usuario conecte desde otro lugar; por contra, si se tratara de un modelo basado en la detección de usos indebidos, mantendrá una lista mucho más amplia que la anterior, pero formada por las direcciones desde las que sabemos, con una alta probabilidad que ese usuario no se va a conectar, de forma que si detectara un acceso

TESIS CON
FALLA DE ORIGEN

desde una de esas máquinas, entonces es cuando el sistema tomará las acciones oportunas.

De cualquier forma, la idea es muy simple: un IDS de tiempo real (los denominados Real Time Intrusion Detection Systems) trabaja continuamente en busca de posibles ataques, mientras que los sistemas que se ejecutan a intervalos (Vulnerability Scanners) son analizadores de vulnerabilidades que cualquier administrador ha de ejecutar regularmente (ya sea de forma manual o automática) contra sus sistemas para verificar que no presentan problemas de seguridad.

Sin importar qué sistemas vigile o su forma de trabajar, cualquier sistema de detección de intrusos ha de cumplir algunas propiedades para poder desarrollar su trabajo correctamente. En primer lugar, y quizás como característica más importante, el IDS ha de ejecutarse continuamente sin que nadie esté obligado a supervisarlo; independientemente de que al detectar un problema se informe a un operador o se lance una respuesta automática, el funcionamiento habitual no debe implicar interacción con un humano. Podemos fijarnos en que esto parece algo evidente: muy pocas empresas estarán dispuestas a contratar a una o varias personas simplemente para analizar las bitácoras o controlar los patrones del tráfico de una red. Sin entrar a juzgar la superioridad de los humanos frente a las máquinas (¿puede un algoritmo determinar perfectamente si un uso del sistema estará correctamente autorizado?) o viceversa (¿será capaz una persona de analizar en tiempo real todo el tráfico que llega a un servidor web mediano?), debemos de tener presente que los sistemas de detección son mecanismos automatizados que se instalan y configuran de forma que su trabajo habitual sea transparente a los operadores del entorno informático. Otra propiedad, y también como una característica a tener siempre en cuenta, es la aceptabilidad o grado de aceptación del IDS; al igual que sucedía con cualquier modelo de autenticación, los mecanismos de

detección de intrusos han de ser aceptables para las personas que trabajan habitualmente en el entorno.

Por ejèmplo, no ha de introducir una sobrecarga considerable en el sistema (si un IDS hace demasiado lenta a una máquina, simplemente no se utilizará) ni generará una cantidad elevada de falsos positivos (detección de intrusiones que realmente no lo son) o de logs, ya que entonces llegará un momento en que nadie se preocupe de comprobar las alertas emitidas por el detector. Por supuesto (y esto puede parecer una tontería, pero es algo que se hace más a menudo de lo que podamos imaginar), si para evitar problemas con las intrusiones simplemente apagamos el equipo o lo desconectamos de la red, tenemos un sistema bastante seguro. . . pero inaceptable.

Una tercera característica a evaluar a la hora de hablar de sistemas de detección de intrusos es la adaptabilidad del mismo a cambios en el entorno de trabajo. Como todos sabemos, ningún sistema informático puede considerarse estático: desde la aplicación más pequeña hasta el propio kernel de Unix, pasando por supuesto por la forma de trabajar de los usuarios (quién nos asegura que ese engorroso procedimiento desde una "desfasada" línea de órdenes mañana no se realizará desde una aplicación gráfica, que realmente hace el mismo trabajo pero que genera unos patrones completamente diferentes en nuestro sistema?), todo cambia con una periodicidad más o menos elevada. Si nuestros mecanismos de detección de intrusos no son capaces de adaptarse rápidamente a esos cambios, están condenados al fracaso. Todo IDS debe además presentar cierta tolerancia a fallos o capacidad de respuesta ante situaciones inesperadas; insistiendo en lo que comentábamos antes sobre el carácter altamente dinámico de un entorno informático, algunos (o muchos) de los cambios que se pueden producir en dicho entorno no son graduales sino bruscos, y un IDS ha de ser capaz de responder siempre adecuadamente ante los mismos. Podemos contemplar, por ejemplo, un reinicio inesperado de varias máquinas o un intento de engaño

hacia el IDS; esto último es especialmente crítico: sólo hemos de pararnos a pensar que si un atacante consigue modificar el comportamiento del sistema de detección y el propio sistema no se da cuenta de ello, la intrusión nunca será notificada, con los dos graves problemas que eso implica: aparte de la intrusión en sí, la falsa sensación de seguridad que produce un IDS que no genera ninguna alarma es un grave inconveniente de cara a lograr sistemas seguros.

Los IDSes basados en la detección de usos indebidos son en principio más robustos que los basados en la detección de anomalías: al conocer la forma de los ataques, es teóricamente extraño que generen falsos positivos (a no ser que se trate de un evento autorizado pero muy similar al patrón de un ataque); es necesario recalcar el matiz "teóricamente", porque como veremos más adelante, la generación de falsos positivos es un problema a la hora de implantar cualquier sistema de detección.

No obstante, en este mismo hecho radica su debilidad: sólo son capaces de detectar lo que conocen, de forma que si alguien nos lanza un ataque desconocido para el IDS éste no nos noticará ningún problema; como ya dijimos, es algo similar a los programas antivirus, y de igual manera que cada cierto tiempo es conveniente (en WINDOWS y derivados) actualizar la versión del antivirus usado, también es conveniente mantener al día la base de datos de los IDSes basados en detección de usos indebidos. Aún así, seremos vulnerables a nuevos ataques. Otro grave problema de los IDSes basados en la detección de usos indebidos es la incapacidad para detectar patrones de ataque. Volviendo al ejemplo de los antivirus, pensemos en un antivirus que base su funcionamiento en la búsqueda de cadenas virales:

Lo que básicamente hará ese programa será buscar cadenas de código hexadécimal pertenecientes a determinados virus en cada uno de los archivos a analizar, de forma que si encuentra alguna de esas cadenas el software asumirá que el fichero está contaminado. Y de la misma forma que un virus

puede ocultar su presencia simplemente cifrando esas cadenas (por ejemplo de forma semialeatoria utilizando eventos del sistema, como el reloj), un atacante puede evitar al sistema de detección de intrusos sin más que insertar espacios en blanco o rotaciones de bits en ciertos patrones del ataque; aunque algunos IDSes son capaces de identificar estas transformaciones en un patrón, otros muchos no lo hacen.

Tras leer la sección anterior seguramente habrá quedado claro que un correcto esquema de detección de intrusos basado en red es vital para proteger cualquier sistema; con frecuencia suele ser el punto más importante, que más ataques detecta, y donde se suelen emplazar la mayoría de sistemas de detección que existen instalados en entornos reales hoy en día. No obstante, esta enorme importancia suele degenerar en un error bastante grave: en muchos entornos los responsables de seguridad, a la hora de trabajar con IDSes, se limitan a instalar diversos sensores de detección basados en red en cada segmento a proteger, creyendo que así son capaces de detectar la mayoría de ataques.

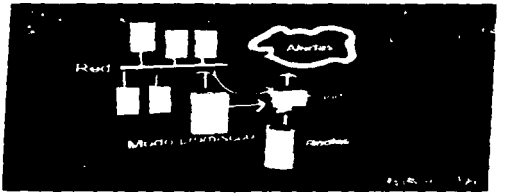
Y eso suele generar una falsa sensación de seguridad, ya que a la hora de lanzar ciertos ataques un pirata puede eludir fácilmente a estos sensores; los sensores de detección en nuestros segmentos de red son importantes, pero no son la panacea. Se puede tener un pirata.

3.4 SNORT

Snort es un IDS en red, está basado en el análisis del tráfico de paquetes, cuenta con un motor de detección de ataques y barrido de puertos que permiten registrar y alertar en tiempo real en caso de ocurrir una intrusión. La detección se basa en la comparación de los patrones de los paquetes capturados contra los patrones residentes en una base de datos que corresponden a ataques. Su arquitectura está principalmente conformada por:

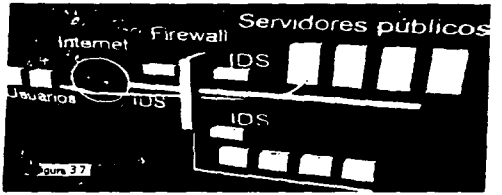
- Decodificador de paquetes
- Un motor de decisiones
- Loggins y alertas

Algo que hace a SNORT interesante es que incorpora un sistema bastante sencillo para escribir nuestras reglas o políticas, a manera de poder adaptarlas



a nuestra aplicación. En la figura anterior se muestra un ambiente con SNORT. Se debe conocer dónde colocar el IDS para su mayor eficiencia; para esto gente dedicada al análisis de redes sugiere instalar los IDS en un punto clave, pareciera que se podría colocar en cualquier punto de la red o en cada tramo; aunque para redes muy grandes se tendría que pensar dos veces el colocar en cada tramo de red, pero algo razonable sería el colocar al IDS en un dispositivo por donde pase todo el tráfico de la red. Si colocamos el IDS detrás del firewall monitorizará todo el tráfico que no sea detectado y parado por el mismo firewall, por lo que será considerado como malicioso en un alto porcentaje de los casos.

La posibilidad de falsas alarmas es muy baja. Al colocarlo antes del firewall capturamos todo el tráfico de entrada y salida de nuestra red, hay una gran posibilidad de falsas alarmas.



En la figura 3.7 se muestra una configuración ambiciosa de IDS con firewall, donde el IDS colocado detrás del firewall juega el papel principal. La colocación de los IDS puede variar de acuerdo a los requerimientos que se tengan; si se trata de una red local sin servicios hacia internet, se suele colocar un solo IDS conectado al tramo de red al que se enganchan los servidores. Si hay conexión a internet, sería muy interesante que el sistema pudiera ver el tráfico desde y hacia Internet.

En una entidad con servicios a internet y con un firewall; cuando se tiene la presencia de un firewall, hay administradores que aprovechan para colocar el IDS en el mismo firewall; así por un lado todo el tráfico hacia y desde Internet pasa por él, por otra parte ahorran costos. Aunque esto es peligroso, porque si logran atacar el firewall, el IDS también se vería comprometido. Si sólo se tiene un IDS lo mejor es colocarlo detrás del firewall, ya que dicho puede filtrar muchos de los ataques.

Aunque los sistemas de detección de intrusos son un buen medio de mantenernos alerta de los ataques al sistema, sería inútil si no tomamos las

precauciones mencionadas en el capítulo I, aspectos como tener buenas contraseñas, firewalls, actualizar aplicaciones y hacer respaldos periódicos.

3.5 HERRAMIENTAS DE SEGURIDAD.

Es importante saber qué herramienta de seguridad implementar; como se ha dicho existe una gran variedad en el mercado informático;

Herramientas genéricas

Son herramientas para administradores, con el fin de proteger redhat o cualquier sistema linux, contra los hackers. Muchas distribuciones de Linux (no solo RedHat) llegan con muchas buenas herramientas de seguridad. Pueden usarlas para mejorar la seguridad de su ordenador. Entre ellas, podemos mencionar; TCPWrappers, PAM (Pluggable Authentication Modules), shadow utilities. Puesto que forman parte de las distribuciones encontrarán mucha documentación.

Las utilidades de shadow permiten la encriptación de passwords. El fichero

/etc/shadow

Remplaza el fichero /etc/passwd. Algo más elaborado es PAM. Como lo dice su nombre, es otro modo de autentificación. PAM permite delimitar el acceso de los usuarios a los servicios. Muchas restricciones pueden ser definidas a partir de ficheros de configuración, facilitando así la administración.

Respecto a los TCPWrappers, permiten reducir el acceso a los servicios de unas máquinas. Pueden ser autorizados o rechazados desde dos ficheros: /etc/hosts.allow y /etc/hosts.deny. Los TCPWrappers pueden ser configurados de dos maneras: sea moviendo los "daemons" o modificando el fichero /etc/inetd.conf.

Para determinar si el cliente tiene permitido conectarse. Luego utiliza el demonio syslog (syslogd) para escribir el nombre del host solicitante y el servicio solicitado a /var/log/secure o /var/log/messages.

Si a un cliente se le permite conectarse, los wrappers TCP liberan el control de la conexión al servicio solicitado y no interfieren más con la comunicación entre el cliente y el servidor.

Además del control de acceso y registro, los wrappers TCP pueden activar comandos para interactuar con el cliente antes de negar o liberar el control de la conexión al servicio solicitado.

Puesto que los wrappers TCP son una utilidad de gran valor a las herramientas de seguridad de cualquier administrador de servidor, la mayoría de los servicios de red dentro de Red Hat Linux están enlazados con la librería libwrap.a. Tales aplicaciones incluyen /usr/sbin/sshd, /usr/sbin/sendmail, y /usr/sbin/xinetd.

Los formatos para /etc/hosts.allow y /etc/hosts.deny son idénticos. Cualquier línea en blanco que comience con un símbolo de numeral o almohadilla (#) será ignorada, y cada regla debe estar en su propia línea.

Las reglas se tienen que formatear de la siguiente manera:

<daemon list>: *<client list>* [: *<option>*: *<option>*: ...]

<daemon list> Es una lista separada por comas de los nombres de procesos (*no* de los nombres de servicios) . La lista de demonios también acepta *operadores* para permitir mayor flexibilidad.

<client list> Es una lista separada por comas de nombres de host, direcciones IP, *patrones* especiales el cual identifica los hosts afectados por la regla. La lista de clientes también acepta operadores.

<option> Es una acción opcional o una lista separada con puntos y comas de acciones realizadas cuando la regla es activada. Se otorga o prohibe el acceso como a continuación se muestra una básica regla de acceso:

vsftpd : informatica.aragon.unam.mx

Esta regla instruye a los wrappers TCP a que vigile conexiones al demonio FTP (vsftpd) desde cualquier host en el dominio informatica.aragon.UNAM.mx. Si esta regla aparece en hosts.allow, la conexión será aceptada. Si esta regla aparece en hosts.deny, la conexión será rechazada.

El próximo ejemplo de regla de acceso es un poco más compleja y utiliza dos campos de opciones:

sshd : informatica.aragon.unam.mx
: spawn /bin/echo `/bin/date` access denied>>/var/log/sshd.log \

: deny

Esta regla de ejemplo indica que si una conexión al demonio SSH (sshd) se intenta desde un host en el dominio informatica.aragon.UNAM.mx, ejecute el comando echo (lo cual registrará el intento a un archivo especial) y rechace la conexión. Puesto que se usa la directiva opcional deny, esta línea rechazará el acceso aún si aparece en el archivo hosts.allow. Los wrappers TCP ofrecen las siguientes ventajas básicas comparado con las otras técnicas de control de servicios de red.

Transparencia tanto para el cliente del host y el servicio de red wrapped. El cliente que se está conectando así como también el servicio de red wrapped no están al tanto de que están en uso los wrappers TCP. Los usuarios legítimos son registrados y conectados al servicio solicitado mientras que las conexiones de clientes prohibidos fallan.

Administración centralizada de protocolo múltiples. Los wrappers TCP operan separadamente de los servicios de red que ellos protegen, permitiendo a muchas aplicaciones de servidor compartir un conjunto común de archivos de configuración para una administración más sencilla.

xinetd

Para controlar el acceso a los servicios de Internet, use xinetd, que es un sustituto seguro del comando inetd. El demonio xinetd conserva los recursos del sistema, proporciona control y registro de acceso, y sirve para arrancar servidores de uso especial. xinetd puede utilizarse para conceder acceso sólo a un grupo de hosts concretos, denegar el acceso a determinados hosts, proporcionar acceso a un servicio en horas concretas, limitar el número de conexiones de entrada y/o la carga que se crea con las conexiones, etc. xinetd se ejecuta de forma permanente y escucha todos los puertos de los servicios que administra. Cuando recibe una petición de conexión de uno de los servicios que administra, xinetd arranca el servidor apropiado a dicho servicio.

El fichero de configuración para xinetd es /etc/xinetd.conf, pero si se examina, se observará que sólo contiene algunos valores por defecto y la instrucción de incluir el directorio /etc/xinetd.d. Para activar o desactivar un servicio xinetd, modifique el fichero de configuración correspondiente del directorio /etc/xinetd.d. Si el atributo disable está definido como yes, el servicio estará desactivado. Si el atributo disable está definido como no, el servicio estará activado.

Firewall

Los Unix libres se liberan con herramientas que permiten transformar su máquina en un firewall. El kernel 2.2 provee "ipchains". El anterior (2.0) usaba "ipfwadm". Para que ipchains o ipfwadm funcionan, el kernel tiene que ser compilado con opciones.

Brevemente, un firewall es una herramienta para filtrar paquetes. Lo más importante concierne su configuración. Es decir, un firewall mal configurado se puede volver muy peligroso. Sin embargo, firewalls son herramientas muy importantes y hay muchos. Por ejemplo, Bastille-Linux provee un firewall basado también en ipchains.

Encriptación

Muchas herramientas forman parte del proceso de encriptación, funcionando en varias áreas. No podremos hablar de todas. Sin embargo, por lo menos, tenemos que decir unas palabras sobre SSH, particularmente la versión libre OpenSSH. La versión actual es 2.3.0. Este producto fue desarrollado primero para funcionar bajo OpenBSD. Hoy, funciona bajo muchos sabores de Unix. OpenSSH reemplaza telnet y los comandos remotos, tales como rsh, rlogin. Incluye scp que remplaza ftp y rcp. OpenSSH permite la encriptación de los datos circulando por la red. Telnet, rsh, etc. transfieren los datos en plano. El problema con estas herramientas viene de las leyes sobre encriptación de los diferentes paises. Las cosas están cambiando, pero en muchos países no se pueden usar libremente éstas herramientas. No obstante, la encriptación es una cosa importante para la seguridad y muchas herramientas tienen que ser tomadas en cuenta:

Proyecto OpenSSL. Librería criptográfica de código abierto que implanta los protocolos SSL/TLS. Los algoritmos implantados no están limitados por las normas de protección contra la exportación

ModSSL. Módulo para convertir el servidor de web más utilizado (Apache) en servidor de web segura con SSL.

Nmap. Escaneador de puertos que permite además determinar el S.O de la máquina remota.Permite escanear servicios TCP, UDP, ICMP, RPC, etc. Es uno de los escaneadores más completos que existen.

Saint. Escaneador de vulnerabilidades bastante completo.

Nessus. Escaneador de vulnerabilidades bastante completo.

Satan. Escaneador de vulnerabilidades bastante completo.

COPS. Chequeador de vulnerabilidades y agujeros de seguridad

Los sistemas de detección de intrusos son potentes mecanismos para mantener de manera segura los sistemas de cómputo y en especial Snort, que a ganado renombre en los últimos días, en el siguiente capítulo se mostrará cómo saber si tenemos un intruso en nuestro sistema.

CAPÍTULO IV
CASO PRÁCTICO DE DETECCIÓN DE INTRUSOS EN LINUX

Leído lo anterior, elegí una de las herramientas para la detección de intrusos; SNORT[18], que es un potente sistema de detección, para tener "segura" nuestra red de posibles ataques, conociendo paso a paso la instalación, configuración y ejecución del IDS. Ahora se ejemplificará un caso real de un servidor de la UNAM comprometido por un hacker, se mostrarán pasos para la detección de intrusos.

La mayoría de los sistemas de detección de intrusos conocidos se basan en encontrar ciertos patrones que impliquen alguna anomalía sobre la red o servidor. Se puede decir que existen los IDS que protegen la red y los IDS que protegen un Host; SNORT puede ser configurado de tres formas (como husmeador, logger y IDS) y para efectos de este trabajo, el sistema sólo se encarga de recabar información de manera local.

La detección de SNORT se basa en la comparación de los patrones de los paquetes capturados contra patrones residentes en una base de datos que corresponden a barridos conocidos. Se puede considerar que SNORT es un IDS en tiempo real. El sistema tiene reglas o políticas de seguridad que realizan chequeos determinados y podemos modificar o crear nuevas reglas de seguridad; podemos nombrar algunas de ellas: backdoor.rules, dos.rules, misc.rules. De esta forma SNORT nos facilita el introducir el puerto destinado al paquete y su contenido, haciendo al sistema bastante flexible.

4.1 INSTALACIÓN Y CONFIGURACIÓN DE SNORT

Comenzaré a dar los pasos para la instalación del SNORT;

SNORT es de distribución gratuita, se puede obtener de la web. El archivo viene en formato RPM, una vez que se tiene el archivo snort-1.8.3-1.i386.rpm en el directorio raiz se procede a instalar, de la siguiente forma; rpm –i snort-1.8.3.rpm

[18] www.snort.org

Los archivos que se generan en Linux, son los mostrados en la siguiente tabla:

/etc/snort.d	/etc/snort.d/classification.config	/etc/snort.d/exploit.rules
/etc/snort.d/attack-responses.rules	/etc/snort.d/ddos.rules	/etc/snort.d/finger.rules
/etc/snort.d/backdoor.rules	/etc/snort.d/dns.rules	/etc/snort.d/ftp.rules
/etc/snort.d/bad-traffic.rules	/etc/snort.d/dos.rules	/etc/snort.d/icmp-info.rules
/etc/snort.d/icmp.rules	/etc/snort.d/scan.rules	/etc/snort.d/web-coldfusion.rules
/etc/snort.d/info.rules	/etc/snort.d/shellcode.rules	/etc/snort.d/web-frontpage.rules
/etc/snort.d/local.rules	/etc/snort.d/smtp.rules	/etc/snort.d/web-iis.rules
/etc/snort.d/log	/etc/snort.d/snort.conf	/etc/snort.d/web-misc.rules
/etc/snort.d/misc.rules	/etc/snort.d/sql.rules	/etc/snort.d/x11.rules
/etc/snort.d/netbios.rules	/etc/snort.d/telnet.rules	/etc/sysconfig/snort
/etc/snort.d/policy.rules	/etc/snort.d/tftp.rules	/usr/bin/snort
/etc/snort.d/porn.rules	/etc/snort.d/virus.rules	/usr/share/doc/snort-1.8.3
/etc/snort.d/rpc.rules	/etc/snort.d/web-attacks.rules	/usr/share/doc/snort-.3/AUTHORS
/etc/snort.d/rservices.rules	/etc/snort.d/web-cgi.rules	/usr/share/doc/snort-1.8.3/BUGS
/usr/share/doc/snort-1.8.3/INSTALL	/usr/share/doc/snort-1.8.3/LICENSE	/usr/share/doc/snort-1.8.3/MIBS
/usr/share/doc/snort-1.8.3/RULES.SAMPLE	/usr/share/doc/snort-1.8.3/SnortUsersManual.pdf	/usr/share/doc/snort-1.8.3/USAGE
/var/log/snort	/usr/share/man/man8/snort.8.gz	

Snort se basa en las librerías de captura de paquetes libcap que provee a Snort de la capacidad de sniffer de paquetes y también las libnet. Sin estas librerías no se puede instalar adecuadamente Snort. Las libnet son utilizadas para múltiples plataformas. A partir de las versiones 7 de Red Hat y Mandrake, la instalación de Snort es muy amigable, ya que no se tienen que estar instalando librerías o parches. Una vez instalado, se procede a configurarlo, existen varias formas, dependiendo la versión que se tenga:

Usando Snort en modo IDS.
Snort –log –dev –h (dirección IP) –c snort.conf

Snort nos informará de intentos de acceso no permitido a nuestro host, así como de escaneos de puertos, ataques DOS, ejecución de exploits, etc. Una de las grandes ventajas de Snort es que nos permite escribir las reglas necesarias para que Snort detecte el ataque, pondré una como ejemplo:

```
alert tcp any any -> 132.248.255.50/74 111 (content:"|00 01 86 a5|";
        msg: "mountd access";)
```

Esta regla detecta el acceso al puerto 111 y saldrá la alerta si el paquete recibido contiene el patrón "000186a5". Afortunadamente, Snort acompaña un buen número de reglas y no necesitamos tener conocimientos de redes para detectar ataques. Estas reglas se instalan en el directorio /usr/local/share/snort.

Para que Snort funcione en modo IDS, debemos pasar el parámetro *-c snort.conf*, siendo snort.conf el fichero de configuración que veremos a continuación.

Lo primero de todo será crear el fichero de configuración snort.conf. Lo que se a hecho es copiar el ejemplo que viene por default y adaptarlo al sistema:

```
# cd /usr/local/etc
# cp snort.conf-sample snort.conf
# cp classification.config-sample classification.config
```

En el fichero de configuración snort.conf es donde especificamos los detalles de nuestra red, los preprocesadores, plugins y donde podemos personalizar nuestras propias reglas. classification.config en cambio clasifica la gravedad de los ataques y le da prioridad a las alertas.

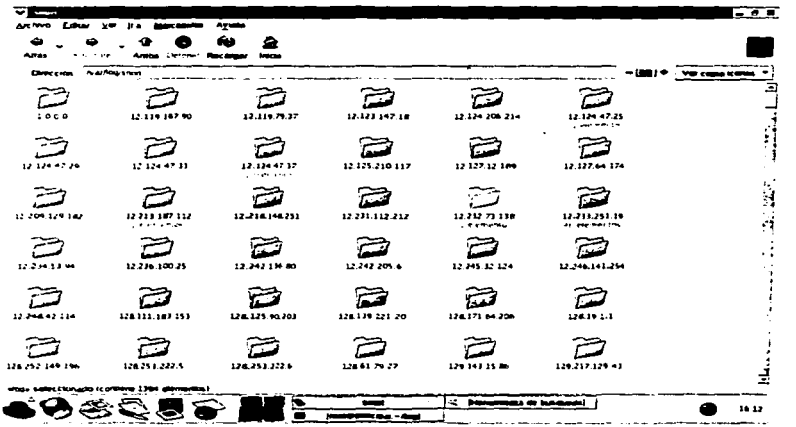
Bien, comencemos por snort.conf. La única modificación que he realizado ha sido especificar la IP de mi red local, en mi caso 132.248.255.50/74:

Una vez realizada la configuración, el siguiente paso a realizar es hacer que snort se ejecute cada vez que arrancamos; para ello he incluido la siguiente linea en el script de arranque */etc/rc.local*:

```
cd /usr/local/share/snort
snort -d -h 192.168.0.1/24 -l /var/log/snort -c /usr/local/etc/snort.conf &
```

De esta forma arrancará cada vez que iniciemos. Una vez hecho esto, para ver quien nos ataca, simplemente vamos al directorio /var/log/snort y miramos los ficheros, en los cuales veremos la hora del ataque, la IP, la fecha, el tipo de ataque que han realizado, y mucha más información , a parte de la estructura de directorios, un archivo alert.ids donde almacenará las alertas generadas. Para crear carpetas por cada IP será necesario añadir la opción -N.

```
SERVCOMU:~# snort -dev -l /var/log/snort -h 132.248.255.50/74 -c
etc/snort/snort.conf -N -D
```



Las alertas en linea de ordenes: completo, rápido, socket, syslog, smb (WinPopup), consola y ninguno, son algunas de las modalidades que Snort ofrece, se explicarán algunas de ellas:

Rápido:

El modo Alerta Rápida nos devolverá información sobre: tiempo, mensaje de la alerta, clasificación , prioridad de la alerta, IP y puerto de origen y destino.

SERVCOMU:~# snort -A fast -dev -l /var/log/snort -h 132.248.255.50/24 -c /etc/snort/snort.conf 09/19-19:06:37.421286

[**] [1:620:2] SCAN Proxy (8080) attempt [**] [Classification: Attempted Information Leak]

Completo:

El modo de Alerta Completa nos devolverá información sobre: tiempo, mensaje de la alerta, clasificación, prioridad de la alerta, IP y puerto de origen/destino e información completa de las cabeceras de los paquetes registrados:

SERVCOMU:~# snort -A full -dev -l /var/log/snort -h 192.168.4.0/24 -c /etc/snort/snort.conf [**] [1:620:2] SCAN Proxy (8080) attempt [**] [Classification: Attempted Information Leak] TTL:128 TOS:0x0 ID:39918 IpLen:20 DgmLen:48 DF ******S* Seq: 0xE87CBBAD Ack: 0x0 Win: 0x4000 TcpLen: 28 TCP Options (4) => MSS: 1456 NOP NOP SackOK Información de la cabecera del paquete: TCP TTL:128 TOS:0x0 ID:39918 IpLen:20 DgmLen:48 DF ******S* Seq: 0xE87CBBAD Ack: 0x0 Win: 0x4000 TcpLen: 28 TCP Options (4) => MSS: 1456 NOP NOP SackOK

Socket:

Manda las alertas a través de un socket, para que las escuche otra aplicación.
SERVCOMU:~# snort -A unsock -c /etc/snort/snort.conf

Veamos dos casos:

Se trata de dos simples accesos a un *servidor proxy ubicado en el puerto 8080 de la máquina destino IP: 132.248.255.55* por parte del *host IP: 192.168.4.3 que realiza la conexión mediante el puerto 1382 en el primer caso y 3159 en el segundo. Snort clasifica o describe esta alerta como un intento de pérdida de información, clasificado como prioridad 2.*

- Modo Alerta Rápida:

```
09/19-19:06:37.421286 [**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2] ...
... {TCP} 192.168.4.3:1382 -> 132.248.255.55:8080
```

- Modo Alerta Completa:

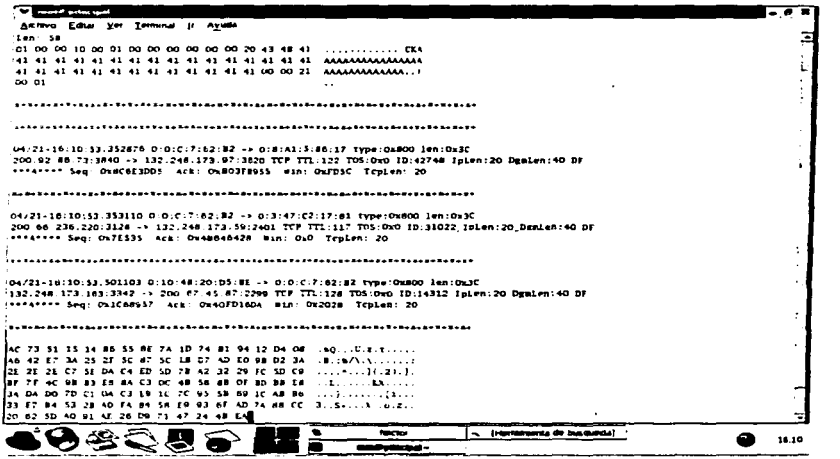
```
[**] [1:620:2] SCAN Proxy (8080) attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
09/19-14:53:38.481065 192.168.4.3:3159 -> 132.248.255.55:8080
TCP TTL:128 TOS:0x0 ID:39918 IpLen:20 DgmLen:48 DF
******S* Seq: 0xE87CBBAD Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1456 NOP NOP SackOK
```

Información de la cabecera del paquete:

```
TCP TTL:128 TOS:0x0 ID:39918 IpLen:20 DgmLen:48 DF
******S* Seq: 0xE87CBBAD Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1456 NOP NOP SackOK
```

Usando Snort en modo sniffer.

Snort –dev



En éste modo se puede visualizar todo el tráfico que pasa en la red; es decir la máquina donde está instalado Snort se coloca en modo promiscuo[19].

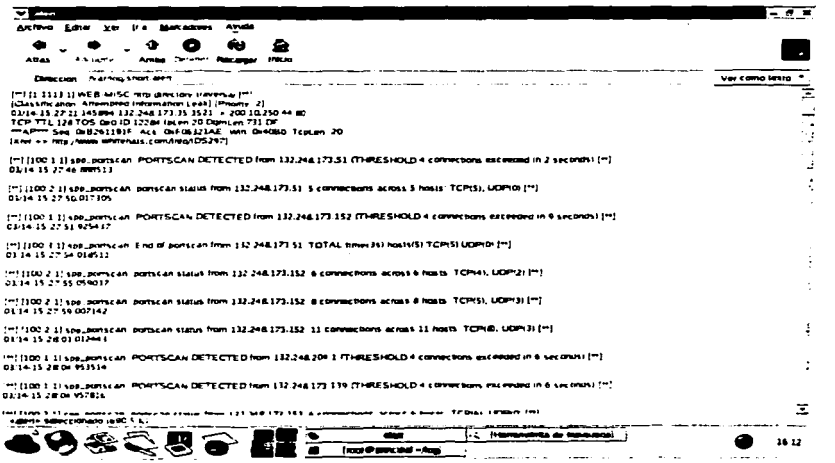
[19] El tráfico de la red se ve sin restricciones

Usando Snort en modo registro de paquetes.

Snort –dev –l log

Dónde la opción -l log nos manda la información a la carpeta log. En esta carpeta se estructurarán una serie de directorios con el nombre de la dirección IP del host que genere el tráfico o intrusión. De la misma manera se hará para un archivo llamado "alert.ids", localizado en la carpeta "log". La opción –dev imprime en pantalla las direcciones IP y cabeceras, los datos que pasan por la interfaz de red con información bastante detallada.

Veamos como funciona en este modo de funcionamiento:



4.2 INTRUSIÓN A UN SERVIDOR LINUX

Ahora que sabemos como configurar Snort, me gustaría mostrar una ejemplificación donde una máquina con linux (version 6.x) se vió comprometida por un intruso, cabe mencionar que es algo real, donde no se tenía un IDS, la máquina era un servidor de web, ftp y correo. Así el intruso logró filtrarse en dicho sistema;

ls –l comando "w" w finger roy finger jesus	El posible intruso hace un listado, pidiendo también se visualicen los permisos y después con el se da cuenta de quienes están conectados al host y de dónde se conectaron. Hace uso del comando "finger" dirigido hacia varios de los usuarios de la máquina.
ls gcc year.c gcc year.c gcc year.c pico>year.c ls pico year.c	En las siguientes líneas; compila un programa en C (year.c) le hace alguna modificación, así como también a bisies.c, no se sabe específicamente el contenido de estos programas, pero se deduce que obtuvo una cuenta
ls pico bisies.c man passwd passwd exit exit ls exit ls	
uname –ar cat /etc/redhat-release exit	En esta parte se tiene el comienzo de código sospecho; usa la instrucción uname, lo cual con los demas parametros "–ar" regresa el hostname. Con el comando "cat" se obtiene la version en la cual se esta trabajando, para posteriormente instalar un exploit que sea acorde a la version del sistema operativo a atacar y despues salir del sistema. Al ingresar el comando"id" , el intruso logra que Linux le muestre las UID reales del sistema, vuelve a checar quien pueda estar observandolo para despues tratar de convertirse en root y salir.
id w su -	

```
exit
cd /dev/sex.ls
w
./v fleur 195 143.8.121  195.143.8.121
w
lynx
```

Se cambia a un directorio llamado sex en /dev y hace un listado, posteriormente usa un navegador de texto para visualizar páginas web, el tener "lynx" activo es un punto de vulnerabilidad, el intruso se conectó a algún sitio en Internet y compila un programa en c y lo modifica, cabe mencionar que el intruso siempre esta alerta de quien pueda darse cuenta de su presencia. Para entonces ya creó un directorio oculto donde guardó algunos programas para después borrarlos. Observa el contenido de las ultimas 10 líneas de "rc.sysinit" , lo modifica y le cambia permisos, para después mostrar la bitácora con "ps aux".

```
gcc -o v2 v.c
pico v.c
gcc -o v2 v.c
w
./v2 fleur 1 1
rm v2.ls
rm v.c
w
tail /etc/rc.d/rc.sysinit
cat ./usr/bin/run
pico /etc/rc.d/rc.sysinit
chmod o-r /etc/rc.d/rc.sysinit
ps aux
exit
w
uname -ar
cat /etc/redhat-release
```

Vuelve a conseguir la versión de linux y el dominio del host que esta comprometiendo, lee las ultimas 10 líneas de passwd y consigue de un sitio web; un exploit. Para que el intruso llegara a este punto, debio haber obtenido la mayor información posible sobre el sistema, modifica el shell baiado y sale.

```
ls
pwd
ps aux
uname -ar
uname -ar
w
strings /bin/login|more
tail /etc/passwd
tail /etc/passwd
wget http://202.189.0.181/exploits/os/linux/redhat/6.1/pamslam.sh
sh pamslam.sh
pico pamslam.sh
```

```
sh pamslam.
ls
rm pamslam.sh
/dev/.crap
exit
```

Al presentar las anteriores ejemplificaciones se pretende que el administrador del sistema, tenga una manera de pensar diferente, ya que conoce como los hackers pueden corromper el sistema.

CONCLUSIONES

Se concluye el trabajo diciendo que la seguridad en un sistema computacional, comienza desde el lugar físico donde se encuentran las computadoras; al construir un centro de cómputo o sala de servidores en una entidad, se debe analizar dónde quedarán ubicados; por lo regular son lugares de difícil acceso u ocultos, también deben hacerse señalizaciones de las areas para personal autorizado y vigilancia. Ya que podemos tener un potente sistema de seguridad y quizá el intruso no pueda accesar a la información de la computadora, pero le es más fácil robar el disco duro de la máquina o incluso llevarse todo el CPU, si no se tiene la seguridad física requerida. Algo que comunmente confunde la gente, es el cómo llamar a los intrusos; es decir si es un hacker o cracker, se dice que el hacker es una persona con ética que sólo se dedica a husmear información pero sin hacer daño alguno; mientras que, el cracker destruye información y no tiene principios morales.

Para prevenir un acceso físico no autorizado a un determinado punto; hay soluciones para todos los gustos, y también de todos los precios: desde analizadores de retina hasta videocámaras, pasando por tarjetas inteligentes o control de las llaves que abren determinada puerta. Todos los modelos de autenticación de usuarios son aplicables, aparte de controlar el acceso lógico a los sistemas, para controlar el acceso físico; de todos ellos, quizás los más adecuados a la seguridad física sean los biométricos.

Teóricamente Linux es seguro; sin embargo, en la vida real varios aprendices de pirata logran "colarse" en los sistemas, el problema radica en las personas que están detrás del sistema operativo, generalmente administradores y usuarios de cualquier categoría. Unix ofrece los mecanismos suficientes como para conseguir un nivel de seguridad más que

aceptable; pero somos nosotros los que en muchos casos no sabemos aprovecharlos. Para solucionar el problema, como ya hemos comentado a lo largo del proyecto, existen dos soluciones que todos deberíamos intentar aplicar: en primer lugar la concienciación de los problemas que nos pueden acarrear los fallos de seguridad (a muchos aún les parece que el tema no va con ellos, que los piratas informáticos sólo existen en el cine, y que en su máquina nada malo puede ocurrir). Tras la concienciación, es necesaria una formación adecuada a cada tipo de persona (evidentemente no podemos exigir los mismos conocimientos a un administrador responsable de varias máquinas que a un usuario que sólo conecta al sistema para lanzar simulaciones); no es necesario convertirse en un experto, simplemente hay que leer un poco y conocer unas normas básicas. Con estos dos pasos seguramente no pararemos a todos los piratas que nos intenten atacar, pero sí a la gran mayoría de ellos, que es lo que realmente interesa en el mundo de la seguridad.

Algunos de las recomendaciones importantes a seguir antes de una intrusión.

Usuarios:

- No elegir claves de menos de seis caracteres, y combinar mayúsculas, minúsculas, números, signos de puntuación, cualquier cosa que nos permita el teclado.

- No apuntar nuestras claves ni compartirlas con otras personas.

- No utilizar nuestra contraseña de acceso en otros sistemas, especialmente juegos en red o equipos Windows.

- Sustituir telnet y ftp por ssh y scp o similares.

- Nunca ejecutar programas que nos envíen por correo o que consigamos a partir de fuentes poco fiable. Tampoco ejecutar ordenes cuyo funcionamiento desconocemos, especialmente si alguien desconocido nos indica teclear "algo" para ver el resultado.

- Desconfiar de llamadas telefónicas o correo electrónico que nos incita a realizar cualquier actividad dentro del sistema, especialmente cambiar nuestra clave; si estas situaciones se producen, indicarlo inmediatamente al responsable de seguridad del equipo, mediante teléfono o en persona.

- Ante cualquier actividad sospechosa que se detecte es recomendable ponerse en contacto con el responsable de seguridad o el administrador, a ser posible por teléfono o en persona.

Como administrador:

- Cerrar los servicios de inetd que no sean estrictamente necesarios.

- No lanzar demonios en el arranque de máquina que no sean estrictamente necesarios.

- Instalar TCP Wrappers y utilizar una política restrictiva.

- Utilizar TCP Wrappers para controlar el acceso a nuestro sendmail propio para este demonio.

- Instalar un sistema Shadow Password para que los usuarios no puedan leer las claves cifradas.

- Deshabilitar las cuentas de usuarios que no conecten al sistema.

- Utilizar versiones actualizadas del núcleo del sistema operativo.

- Pasar un crakeador de password en tu sistema para ver si son fiables los password de los usuarios.

- Generar los password de manera aleatoria; por el administrador o por algún programa.

Una de las razones por la que esta tesis se encaminó hacia Linux fué el que es un sistema operativo del medio académico que constantemente tiene cambios, es de distribución gratuita y bueno, el análisis de la detección de intrusos se tenía que llevar a cabo en un sistema operativo en específico. La elección de Snort como sistema de detección de intrusos es para mejorar la seguridad en el sistema, porque comparado con otras herramientas igual de eficientes, Snort es de distribución gratuita, se puede mudar a otra plataforma como Windows y es muy flexible. Se han mencionado los diferentes tipos de IDS y la realidad en el medio de la computación arroja que los IDS basados en red están cobrando un auge importante en la computación sin descartar a los de host (honeypots). Cabe mencionar que la forma en que Snort hace la detección es una técnica muy poderosa que para efectos de este trabajo no se referenciará; se están desarrollando sistemas equivalentes, cubriendo las imperfecciones que se tienen. El lugar dónde se instalan los IDS es según las necesidades requeridas, así como los recursos económicos que se tengan. En ésta tesis se mencionaron algunas configuraciones, para la implantación de un IDS, se deben tomar en cuenta los puntos mencionados; el IDS por lo general es colocado donde hay un mayor flujo de información, pero aún así; una computadora no está 100% segura, así que tenemos que tener en cuenta que se puede hacer después de que un intruso logra entrar al sistema.

Después que un intruso haya entrado al sistema, ya se tiene que actuar sin cometer ningún fallo, sin dejarlo pasar en el tiempo y sobre todo lo más rápido posible; esto dependiendo las políticas que se tengan, ya que cuando se tiene un tarro de miel, se requiere que el intruso permanezca suficiente tiempo para analizar su ataque. Un administrador tendría diferentes opciones;

1.-Apagaría el sistema.
2.-Quitaría el cable que tiene conectado a la red.
3.-Estudiaríá el comportamiento del intruso para saber las debilidades del sistema.

La mejor elección sería la 3, pero sin tomar demasiados riesgos de pérdida de información.

Un hacker puede tardar meses en vulnerar un sistema ya que son cada vez más sofisticados. Pero el lema es viejo: hecha la ley, hecha la trampa. Se dice que una computadora puede estar segura sólo estando apagada, pero no es eficiente, también se dice que la mejor forma de llevar a cabo una bitácora es la escrita a mano, ya que los intrusos pueden modificar las que están en el sistema.

Ahora quiero mencionar el hecho que en México no existen el tipo de leyes como en otros países, un abogado en este país podría interpretar ciertas leyes para tomar acciones contra un intruso informático, tales como estafa.

Con esta aproximación a un tema de gran interés y de preocupación, también se puede señalar que dado el caracter transnacional de los delitos cometidos mediante el uso de las computadoras; es conveniente, establecer tratados de extradición o acuerdos de ayuda mutua entre los países, que permitan fijar mecanismos sincronizados para la puesta en vigor de

instrumentos de cooperación internacional para contrarrestar eficazmente la incidencia de la criminalidad informática.

Asimismo, la problemática jurídica de los sistemas informáticos debe considerar la tecnología de la información en su conjunto (chips, inteligencia artificial, nanotecnología, redes, etc.), evitando que la norma jurídica quede desfasada del contexto en el cual se debe aplicar.

Por otro lado, se observa el gran potencial de la actividad informática como medio de investigación, especialmente debido a la ausencia de elementos probatorios que permitan la detección de los ilícitos que se cometan mediante el uso de los ordenadores. Finalmente, debe destacarse el papel del Estado, que aparece como el principal regulador de la actividad de control del flujo informativo a través de las redes informáticas.

GLOSARIO

A

Aging Password: Envejecimiento de contraseñas.
Anomaly Detection: Detección de anomalías.
Availability: Disponibilidad.

B

Back Door: Puerta trasera.
Backup: Copia de seguridad.
Backup level: Nivel de copia de seguridad.
Bug: Agujero.

C

Copyright: Derechos reservados de autor.
CPU: Unidad de procesamiento central.
Cryptoperiod: Tiempo de expiración de clave, vigencia de clave.

D

Demonio: Programa encargado de brindar un servicio.
DES: Algoritmo de encriptación estándar.

E

Eavesdropping: Fisgoneo, interceptación.
Entrapment: Trampeado.
Exploit: Programa que compromete un sistema.

F

Fault: Fallo.

Firewall: Cortafuegos.

G

Group Identifier: Identificador de grupo (GID).

H

Honeypot: Tarro de miel, sistema de decepción.

Host authentication: Autenticación por máquina.

Host{based IDS: Sistema de detección de intrusos basado en máquina.

I

Integrity: Integridad.

Intrusion Detection System: Sistema de detección de intrusos (IDS).

L

Log File Monitor: Monitor de registros (LFM).

M

Malware: Software malicioso.

Mandatory Access Control: Control de accesos obligatorio (MAC).

Misuse Detection: Detección de usos indebidos.

Multilevel security: Seguridad multinivel (MLS).

N

Network based IDS: Sistema de detección de intrusos basado en red.

Notarization: Certificación.

O

One Time Password: Clave de un solo uso, clave de uso único .

P

Passive Wiretapping: Fisgoneo, interceptación.

Password: Clave, contraseña.

Patch: Parche.

Pattern Matching: Comparación y emparejamiento de patrones.

Personal Identification Number: Número de identificación personal (PIN).

Privacy: Privacidad.

R

Rom: Memoria de sólo léctura

Ram:Memoria de acceso aleatorio.

S

Safety: Seguridad (entendida como tolerancia a fallos).

Scavenging: Basureo.

Security policy: Política de seguridad.

Security: Seguridad.

Shadow Password: Oscurecimiento de contraseñas.

Snooping: Fisgoneo.

T

Threat: Amenaza.

Trap Door: Puerta Trasera.

Trashing: Basureo.

Trojan Horse: Caballo de Troya.

Trojan Mule: Mula de Troya.

TCP Wrappers: Programa de seguridad en Unís.

U

Uninterruptible Power Supplies (UPS): Servicio de Alimentación Ininterrumpido (SAI).

User Identifier: Identificador de usuario (UID).

W

Wiretapping: Interceptación.

Worm: Gusano.

BIBLIOGRAFÍA

www.cert.org

www.gnu.org

www.redhat.com

[1].Simson Garfinkel
Gene Spafford
Practical Unix and Internet Security
McGRAW-HILL
USA
Junio 1999

[2].Rebecca Gurley Bace
Intrusion Detection
Technology Series
USA
Enero 1998

[3].James Mohr
Linux
Prentice hall hispanoamericana, s.a
Mexico 1999

[4].Stuart McClure, Joel Scambray, George Kurtz
Hacking Exposed: Network Security Secrets and Solutions
Computing McGraw-Hill
USA September 10, 1999