



FACULTAD INFORMÁTICA Y CIENCIAS APLICADAS
CARRERA TÉCNICO EN INGENIERÍA EN REDES COMPUTACIONALES



TEMA:

Manual técnico de aplicaciones de estrategias de seguridad para minimizar las vulnerabilidades de Linux a nivel de servidor de servicio de Internet con Red Hat 9.0 para ser utilizado por los alumnos de la carrera Técnico en ingeniería de Redes Computacionales.

Trabajo de Graduación presentado por:

Luis Edgardo Pérez Mejía

Sabino Mendoza Lázaro

Franklin Alberto Zelada

Para Optar Al grado de:

TÉCNICO EN INGENIERÍA DE REDES COMPUTACIONALES

MARZO, 2008

SAN SALVADOR, EL SALVADOR, CENTROAMÉRICA.

PAGINA DE AUTORIDADES

LIC. JOSÉ MAURICIO LOUCÉL

RECTOR

ING. NELSON ZÁRATE SÁNCHEZ

VICARRECTOR ACADEMICO

ING. LORENA DUQUÉ DE RODRÍGUEZ

DECANO

JURADO EXAMINADOR

LIC. MARVIN HERNANDEZ

PRESIDENTE

ING. VICENTE ANTONIO ZARCEÑO

PRIMER VOCAL

ING. ERNESTO EMÉSTICA

SEGUNDO VOCAL

MARZO, 2008

SAN SALVADOR, ELSALVADOR, CENTROAMÉRICA.

INDICE

Introducción _____ *i*

CAPITULO I GENERALIDADES DEL PROYECTO TEMÁTICO

1.1 Situación problemática	1
1.2 Enunciado del problema	2
1.3 Justificación	2
1.4 Objetivos	3
1.5 Alcances	4
1.6 Delimitaciones	4
1.7 Estudios de factibilidad	5

CAPITULO II MARCO TEÓRICO DE REFERENCIA

2.1 Elaboración de un manual técnico	12
2.2 Breve historia de Internet y generalidades	18
2.3 Los antecedentes históricos de Linux	23
2.4 Tipos de ataques al Sistema Operativo de Red	
2.4.1 Ingeniería social	25
2.4.2 Programas caballos de Troya	29
2.4.3 Archivo tar malignos	35
2.4.4 Paquetes rpm corruptos	39
2.4.5 Virus y gusanos	44

2.5 Ataques desde la Red	
2.5.1 Rutas de confianza y programas troyanos	47
2.5.2 Sudo	48
2.5.3 Programas sxid	54
2.5.4 Ataques por desbordamientos de búfer contra ejecutables suid de root: Sudo	54
2.5.5 Ataques al núcleo del sistema	58
2.6 Algunas medidas de seguridad al servidor en Linux Red Hat	
2.6.1 Iptables	64
2.6.2 Los wrappers TCP y el comando xinetd	74

CAPITULO III PROYECTO TEMÁTICO

3. 1Planteamiento del proyecto temático	92
3.2 Cronograma del proyecto	94
3.3 Documentación Técnica	95
3.4 Evaluación Técnica y Económica	105
3.5 Tecnologías y Recursos Seleccionados	108
3.6 Presupuesto proyectado	109
3.7 Oferta Técnica	110
3.8 Oferta Económica	111

CAPITULO IV PROPUESTA

4.1 Artículo de manual técnico de estrategias de seguridad para servidores de servicios de Internet con Linux Red Hat 9.0	112
4.2 Presentación del Prototipo (Power Point)	114
4.3 Configuraciones de seguridad para el servidor	119
Conclusión	131
Recomendación	132
Bibliografía	133
Anexos	134

INTRODUCCIÓN

En los últimos años se ha observado un gran avance en el sistema operativo Linux Red Hat 9.0, los cuales han traído un gran beneficio a los usuarios que lo utilizan, pero también se han incrementado el número de personas que intentan sacar provecho ilegalmente buscando vulnerabilidades de este sistema operativo para poder conseguir sus propósitos tales como extraer información valiosa de los usuarios también utilizar su estación de trabajo para hacer ataques a un servidor específico o inutilizar un C.P.U. solo por diversión, lo cual también trae costos importantes para los usuarios de este sistema operativo.

Es así a creación de este manual técnico, para beneficio de los estudiantes de la Carrera Técnico en Ingeniería en Redes Computacionales, los cuales tendrán que lidiar con este tipo de inconvenientes que se generan en este sistema de esta forma obtendrá un apoyo para conocer las deficiencias y así poder minimizar sus vulnerabilidades.

CAPITULO I

GENERALIDADES DEL PROYECTO

1.1 SITUACIÓN PROBLEMÁTICA

En la actualidad la mayoría de redes informáticas trabajan bajo la plataforma de Microsoft, dejando como segunda alternativa otros sistemas operativos que aunque son muy utilizados, no se conocen en el ambiente laboral, en algunos casos se utilizan para aplicaciones específicas como lo es en servidores, dentro de estos sistemas operativos se encuentra Linux con todas sus distribuciones las cuales tienen campos de aplicaciones específicas y que son consideradas más estables y sujetas a un menor número de ataques.

El sistema operativo Linux Red Hat 9.0 se considera seguro, robusto con menos vulnerabilidad que otros sistemas operativos, esto es debido al poco conocimiento que se tiene sobre el sistema y a la poca implementación que se le da en nuestro medio, pero al profundizar y estudiarlo más se encuentra que posee ciertas debilidades y deficiencias, que pueden afectar directamente el rendimiento y la seguridad de una red, así como las aplicaciones que se ejecutan en ella, tal es el caso de un servidor de servicio de Internet que está operando con Linux Red Hat 9.0

Otro factor a considerar es la falta de documentos que contengan información relacionada con Medidas de Seguridad que deben de implementarse en aplicaciones como es el caso de Servidores de Servicio de

Internet trabajando bajo plataforma de Linux Red hat 9.0, que especifica sobre algunas medidas de seguridad a nivel de Servidores de Servicio de Internet.

1. 2 ENUNCIADO DEL PROBLEMA

¿De que forma se puede contribuir con los estudiantes de la Carrera de Técnicos en Ingeniería de Redes de la UTEC, a proporcionar información que sirva de apoyo para minimizar las vulnerabilidades que presenta Linux Red Hat 9.0 a nivel de servidores de Servicio de Internet ?

1.3 JUSTIFICACIÓN

Desarrollar un manual técnico que contenga información relacionada a vulnerabilidades y seguridad a nivel de Linux Red Hat 9.0 en ambiente de servidores de Servicio de Internet, basados bajo los siguientes criterios:

- Documentar y describir algunas de las vulnerabilidades que posee el sistema operativo Linux Red Hat 9.0
- Proporcionar los procedimientos para reforzar la seguridad en el sistema operativo Linux Red Hat 9.0
- Describir algunos métodos de configuración para brindar mayor seguridad en el sistema operativo Linux Red Hat 9.0

1.4 OBJETIVO GENERAL

Facilitar el aprendizaje a los alumnos de la carrera Técnico en Ingeniería en Redes Computacionales, en lo relacionado a la aplicación de estrategias de seguridad en el sistema operativo Linux Red Hat 9.0, a través de un manual técnico que contendrá algunas configuraciones básicas de servidores de servicio de Internet.

- **Objetivos Específicos**

Presentar los elementos que alteran las vulnerabilidades en el sistema operativo Linux Red Hat 9.0 aplicado a un servidor de servicio de Internet.

Listar los mecanismos necesarios que permitan contrarrestar las vulnerabilidades de los servidores de servicio de Internet basados en Linux Red Hat 9.0

Elaborar un manual técnico sobre procedimientos de configuración para aumentar la seguridad en los servidores de servicio de Internet en Linux Red Hat 9.0.

1.5 ALCANCES

Entregar un manual que contenga algunas de las vulnerabilidades que presenta el sistema operativo Linux Red Hat 9.0, en un ambiente de servidores de servicio de Internet.

El manual contendrá procedimientos de configuración tales como: xinetd, iptables, entre otros, para aumentar la seguridad en los servidores, utilizando el sistema Linux Red Hat 9.0.

1.6 DELIMITACIONES

- **Geográfica:**

Edificio Francisco Morazán de la Universidad Tecnológica de El Salvador, ubicada en calle Arce nº 1020, San Salvador.

- **Temporal:**

El manual técnico de estrategias de seguridad para minimizar las vulnerabilidades a nivel de servidor de Servicio de Internet en Red Hat 9.0, tendrá una fecha de elaboración que comprenderá d el mes de Agosto hasta Diciembre del 2007

- **Especifica:**

El manual estará elaborado en formato PDF y estará disponible en un CD -R.

1.7 ESTUDIOS DE FACTIBILIDAD TÉCNICA

La propuesta, en la elaboración de un manual técnico de aplicación de estrategias de seguridad en servidores de Servicio de Internet con Linux Red Hat 9.0 para el laboratorio de Redes de la Universidad Tecnológica de El Salvador, requiere de lo siguiente:

- Adquisición del Software Linux Red Hat 9.0 en CD.

- Adquisición de fotocopias de extractos del Libro “Hackers en Linux” 2ª parte.
- Impresiones de manuales profesional para utilización de Linux Red Hat 9.0.
- Uso de la herramienta Internet para la búsqueda de información sobre Linux Red Hat 9.0.
- Utilización de equipos (computadoras), para el desarrollo, comprobación y pruebas técnicas para la elaboración del manual.

Por lo tanto; después de corroborar las herramientas y tecnologías disponibles, se observa que existen los elementos necesarios para la elaboración del manual.

- **Económica**

RECURSOS EXISTENTES		
CANTIDAD	DESCRIPCIÓN	COSTOS
800	Fotocopias (hojas) a un costo (0.2 cts.)	\$16.00
	Transporte	\$300.00

45 horas	Internet, a un costo (\$ 1.00)	\$45.00
400	Impresiones, a un costo (\$0.07)	\$28.00
3	Computadoras	Propias
Total		\$ 389.00

Tomando en cuenta, lo ya planteado anteriormente, se confirma que se cuenta con los recursos económicos necesarios para la creación del manual técnico de estrategias de seguridad de servidor de servicio de Internet.

- **Operativa**

El manual estará orientado hacia los alumnos de la carrera Técnico en Ingeniería de Redes Computacionales que cursan las materias de Sistemas Operativos y Administración de Redes, los cuales tendrán la posibilidad de utilizarlo y a su vez se verán beneficiados con las aplicaciones y estrategias de seguridad de Linux Red Hat 9.0 que contendrá el manual.

Se consultó con los alumnos de algunas asignaturas de la carrera Técnico en Ingeniería de Redes Computacionales para recopilar información sobre la aceptación de un manual técnico de aplicaciones y estrategias de seguridad

para los servidores de servicio de Internet con el sistema operativo Linux Red Hat 9.0, el estudio se hizo a través de una encuesta .

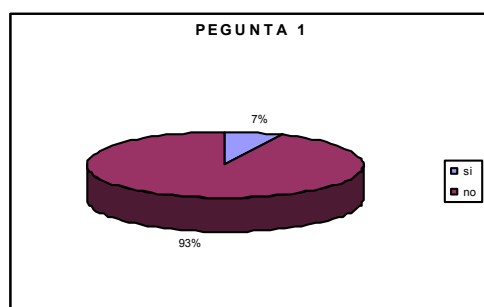
- **Encuesta realizada en la Universidad Tecnológica de El Salvador**

La presente encuesta tiene como finalidad, recopilar información para medir el conocimiento, la capacidad de aplicación y aceptación de un manual técnico de estrategias de seguridad para minimizar las vulnerabilidades de Linux a nivel de servidores de servicios de Internet con Red Hat 9.0; para los estudiantes de la carrera, Técnico en Ingeniería de Redes Computacionales.

1-¿En lo que va de su carrera de Técnico en Ingeniería de Redes Computacionales ha recibido enseñanza didáctica sobre el sistema operativo Linux?

Personas que respondieron:

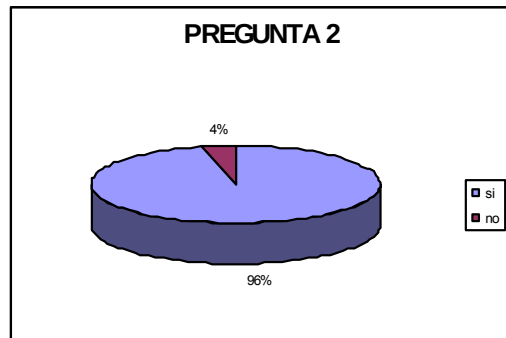
SI	NO
7	43



2- ¿Conoce algún tipo de estrategia de seguridad para la prevención de ataques en Servidores de servicios de Internet con Linux Red Hat 9.0?

Personas que respondieron:

SI	NO
4	46



3- ¿Le gustaría conocer algunas estrategias de aplicación que le puedan proporcionar algún nivel de seguridad en servidor de servicios de Internet con Linux Red Hat 9.0?

Personas que respondieron:

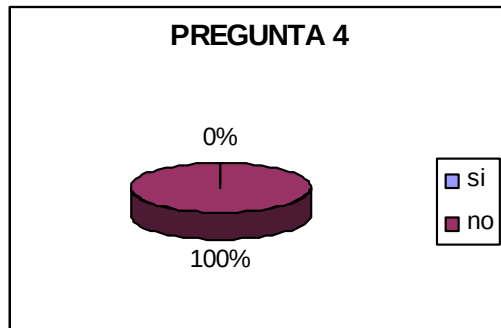
SI	NO
50	0



4- ¿Tiene conocimiento usted, si el laboratorio consta de un material que sirva de apoyo para la prevención de ataques en servidores de servicios de Internet con Linux Red Hat?

Personas que respondieron:

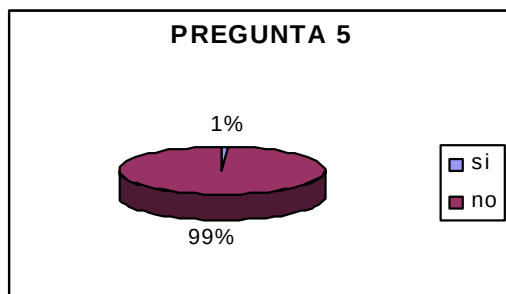
SI	NO
0	50



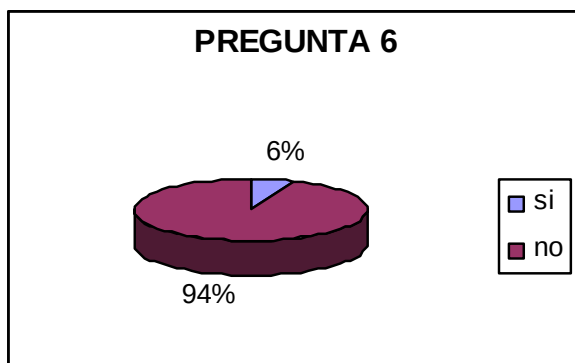
5- ¿Considera usted, como estudiante de la UTEC conveniente; la creación de un manual técnico que contenga aplicaciones y estrategias de seguridad para prevenir los ataques de los servidores de servicios de Internet con Linux Red Hat 9.0?

Personas que respondieron:

SI	NO
1	49



6- ¿Ha desarrollado usted, practicas en el Laboratorio de Redes ha cerca del Sistema Operativo Linux Red Hat 9.0?



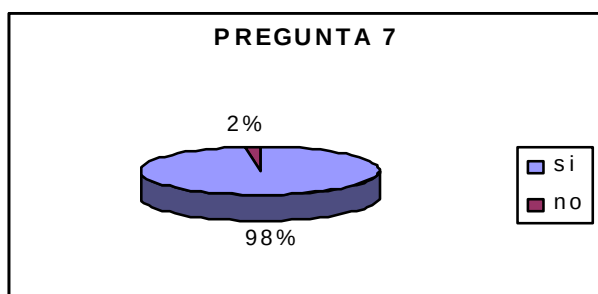
Personas que respondieron:

SI	NO
6	44

7- ¿Considera usted necesario, el uso de un manual técnico de aplicaciones que contenga algunas estrategias de seguridad para prevenir los ataque de los servidores de servicios de Internet con Linux Red Hat 9.0?

Personas que respondieron:

SI	NO
48	2



CAPITULO II MARCO TEÓRICO DE REFERENCIA

2.1 ELABORACIÓN DE UN MANUAL TÉCNICO

Este documento contiene toda la información sobre los recursos utilizados por el proyecto, llevan una descripción muy bien detallada sobre las características físicas y técnicas de cada elemento. Por ejemplo: características de procesadores, velocidad, dimensiones del equipo, garantías, soporte, proveedores y equipo adicional.

Su extensión depende de la cantidad de recursos y equipo utilizado y generalmente se presenta en forma de fichas técnicas en donde se describe en cada una las características de cada recurso.

o CONSIDERACIONES GENERALES PARA LA DOCUMENTACIÓN DE EL DESARROLLO DE APLICACIONES INFORMATICAS:

Toda documentación que se genere para un proyecto específico, que haya sido revisada y aprobada, debe poseer lo siguiente:

a) Identificación del documento .

Este debe poseer la siguiente información:

- Logotipo de la organización.

- Nombre oficial de la organización.
- Denominación y extensión. (De corresponder a una unidad en particular debe anotarse el nombre de la misma.)
- Lugar y fecha de elaboración.
- Número de revisión (en su caso).
- Unidades responsables de su elaboración, revisión y/o autorización.
- Clave de la forma. Primero, las siglas de la organización; segundo las siglas de la unidad administrativa donde se utiliza la forma y, por último, el número de la forma. Entre las siglas y el número debe colocarse un guión o diagonal. (En su caso)

b) Estructura del documento.

Por cada documento final deberá entregarse copias al personal involucrado en el proyecto.

Una vez concluido el desarrollo del manual, (considerando para esto los posibles cambios que se efectúen durante la etapa de garantía de que lo cubre (si así fuera el caso)), el usuario final debe recibir una versión actualizada del manual técnico.

o ESTRUCTURA DEL DOCUMENTO MANUAL TÉCNICO

Índice. Relación de los capítulos y páginas correspondientes que forman parte del documento

Introducción. Se debe presentar una breve descripción del sistema desarrollado, que contemple el ámbito abarcado, cual es su función principal y un detalle de las funciones macros o partes que lo componen. Puede incluir un mensaje de la máxima autoridad de las áreas comprendidas en el manual.

Objetivo general del sistema. Se debe describir el objetivo general del sistema.

Objetivos específicos. Se deben describir brevemente los objetivos específicos que se cumplieron con el desarrollo del sistema.

Contenido técnico. Definición de reglas del negocio implementadas en el sistema desarrollado.

Diagramas de flujo de datos. Junto con su respectivo diccionario de datos:

- Controles de auditoria implementados en el sistema.
- Descripción de campos requeridos por pantalla con presentación de pantallas.
- Diagrama de navegación del sistema.
- Requerimientos de interfase con otros sistemas.

- Modelo lógico de datos, diagrama entidad -relación.
- Modelo de datos físico, junto con su respectivo diccionario de datos.
- Matriz de procesos versus organización.
- Matriz de programas versus entidades.

Plataforma de usuario. Aquí se describen los requerimientos mínimos que se deben tener tanto de hardware como de software para que el sistema se pueda instalar y ejecutar correctamente (en caso de que se considere necesario).

Áreas de aplicación y/o alcance de los procedimientos . Esfera de acción que cubren los procedimientos

Responsables. Para iniciar los trabajos que conducen a la integración de un manual, es indispensable prever que no queda diluida la responsabilidad de la conducción de las acciones en diversas personas, sino que debe designarse a un coordinador, auxiliado por un equipo técnico, al que se le debe encomendar la conducción del proyecto en sus fases de diseño, implantación y actualización. De esta manera se logra homogeneidad en el contenido y presentación de la información. Por lo que respecta a las características del equipo técnico, es conveniente que sea personal con un buen manejo de las relaciones humanas y que conozca a la organización en lo que concierne a sus objetivos, estructura, funciones y personal. Para este

tipo de trabajo, una organización puede nombrar a la persona que tenga los conocimientos y la experiencia necesarios para llevarlo a cabo. Por la naturaleza de sus funciones puede encargarlo al titular del área específica. Asimismo, puede contratar los servicios de consultores externos.

o ¿QUÉ ES SEGURIDAD?

Seguridad es toda aquella acción que tiende a garantizar el cumplimiento de cuatro objetivos importantes:

Confidencialidad: Los objetos de un sistema serán accedidos únicamente por entidades autorizadas.

Integridad: Los objetos sólo pueden ser modificados por entidades autorizadas y de manera controlada.

Disponibilidad: Los objetos deben permanecer accesibles a entidades autorizadas.

Autenticación: Verificar la identidad del emisor y del receptor.

Los tres elementos que se deben proteger son: el hardware (servidor, cableado, etc.), el software (sistema operativo, aplicaciones, etc.) y los datos (documentos, bases de datos, etc.). Las amenazas a las que pueden estar expuestos estos elementos son:

Interrupción: Un recurso del sistema es destruido o se vuelve no disponible. También es denominado *Negación de Servicio*.

Intercepción: Una entidad no autorizada consigue acceso a un recurso.

Modificación: Una entidad no autorizada no solo consigue acceder a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad.

Fabricación: Una entidad no autorizada inserta objetos falsificados en el sistema.

2.2 BREVE HISTORIA DE INTERNET

Los primeros conceptos acerca de la red se desarrollaron en el año 1973, realizándose las primeras pruebas de interconexión de redes en julio de 1977. Se puede considerar que Internet ya estaba en actividad en los Estados Unidos, alrededor de 1982 y a finales de la década de los 80 comienza a expandirse internacionalmente, incluyendo usuarios y redes de distintas partes del mundo.

Sin embargo, hasta alrededor del año 1993, el uso de Internet estaba, en su mayor parte, limitado a círculos técnicos, científicos y académicos. La gran mayoría de la población, incluidas personas familiarizadas con la informática y el uso de ordenadores, nunca habían oído hablar de Internet. En

determinado momento se produce un punto de inflexión en el cual todos los medios de difusión comienzan a hablar de Internet, el gran público empieza a interesarse por el tema, la Red comienza a insertarse en los distintos ámbitos de la sociedad y a tener implicaciones económicas importantes. Surge World Wide Web, la telaraña mundial.

- **SERVICIO DE PROXY**

Los proxies son máquinas que tienen como misión distribuir el tráfico en la red, de tal manera que las conexiones que se solicitan desde un ordenador local a Internet, pueden dirigirse hacia un servidor o hacia otro según la carga solicitada.

- **SERVIDOR (SERVER):**

El servidor es la máquina principal de la red, la que se encarga de administrar los recursos de la red y el flujo de la información. Muchos de los servidores son "dedicados", es decir, están realizando tareas específicas, por ejemplo, un servidor de impresión solo para imprimir; un servidor de comunicaciones, sólo para controlar el flujo de los datos...etc. Para que una máquina sea un servidor, es necesario que sea una computadora de alto rendimiento en cuanto a velocidad y procesamiento, y gran capacidad en disco duro u otros medios de almacenamiento.

- **ESTACIÓN DE TRABAJO (WORKSTATION):**

Es una computadora que se encuentra conectada físicamente al servidor por medio de algún tipo de cable. Muchas de las veces esta computadora ejecuta su propio sistema operativo y ya dentro, se añade al ambiente de la red.

- **SISTEMA OPERATIVO DE RED:**

Es el sistema (Software) que se encarga de administrar y controlar en forma general la red. Para esto tiene que ser un Sistema Operativo Multiusuario, como por ejemplo: Unix, Netware de Novell, Windows NT, etc.

- **¿QUÉ ES UN PROTOCOLO?**

Los protocolos de comunicación definen las normas que posibilitan que se establezca una comunicación entre varios equipos o dispositivos, ya que estos equipos pueden ser diferentes entre sí. Un interfaz, sin embargo, es el encargado de la conexión física entre los equipos, definiendo las normas para las características eléctricas y mecánicas de la conexión.

Todos los protocolos y estándares que se consolidan como propios de Internet han de ser organizados y dirigidos de alguna manera. Esta es la misión principal del IETF (*Internet Engineering Task Force*), que es una gran comunidad de carácter abierto formada por diseñadores de redes, operadores, usuarios, etc. Todos los protocolos agrupados normalmente bajo el nombre TCP/IP son estándares de Internet cuyo desarrollo depende del

IETF. Las actividades que realiza el IETF se dividen en distintos grupos, llamados *Working Groups* (WG) con finalidades específicas, los cuales se clasifican en distintas áreas comunes (Aplicaciones, seguridad, estandarización, servicios de transporte, etc.). El IESG (*Internet Engineering Steering Group*) se encarga de coordinar y dirigir al IETF por medio de los directores de área, que controlan las actividades número de los *Working Groups* que se encuentren dentro de cada área.

- **¿QUÉ ES TCP/IP?**

El Internet es una red de ordenadores conectados juntos a través de redes de comunicaciones. Esta red consiste en enlaces de fibra óptica, satélite, radio y las líneas telefónicas. El sistema tiene ordenadores de todos los tipos y funcionamiento todo el tipo de sistemas operativos. Todos utilizan el **TCP/IP** como lenguaje común.

Cuando se habla de TCP/IP, se relaciona automáticamente como el protocolo sobre el que funciona la red Internet. Esto, en cierta forma es cierto, ya que se le llama TCP/IP, a la familia de protocolos que nos permite estar conectados a la red Internet. Este nombre viene dado por los dos protocolos estrella de esta familia:

- El **Protocolo TCP**, funciona en el nivel de transporte del modelo de referencia OSI, proporcionando un transporte fiable de datos.

- El **Protocolo IP**, funciona en el nivel de red del modelo OSI, que nos permite encaminar nuestros datos hacia otras maquinas.
- **KERBEROS.**

Kerberos es un protocolo de seguridad de amplio soporte que utiliza un dispositivo especial conocido como servidor de autenticación. Este revalida contraseñas y esquemas de encriptado. Kerberos es uno de los sistemas de encriptamiento más seguros utilizados en comunicaciones.

- **SERVIDOR DE NOMBRE DE DOMINIO .**

El servidor de nombre de dominio (Domain Name Server, DNS) habilita un dispositivo con un nombre común para que sea convertido a una dirección especial de red. Por ejemplo, no se puede tener acceso a un sistema llamado daniel_laptop desde una red del otro lado del país, al menos que éste disponible algún método de verificación de los nombres de las máquinas locales. DNS proporciona la conversión del nombre común local a la dirección física única de la conexión de red del dispositivo.

- **SECURE SHELL (SSH)**

Protege conexiones TCP mediante criptografía (a nivel de OSI de presentación, no a nivel de transporte o inferiores). Se protege, por tanto, conexión a conexión.

Generalmente nos imaginamos a los crackers como personas muy pálidas, tímidas, inadaptadas socialmente, orfos de los ordenadores, gente que a las res de la mañana siguen analizando miles de líneas de códigos buscando fallos que les permitan lanzar sus ataques, que consisten infiltrarse en ordenadores conectados a Internet través de sus módems y que realizan toda su actividad ilegal sin ningún tipo de contacto con otros humanos. Sin embargo, los crackers realmente peligrosos hacen uso de técnicas encaminadas a conseguir que sea usted el que le haga todo el trabajo sucio.

2.3 HISTORIA DE LINUX

o EL PADRE DE LINUX: LINUS TORVALDS

Linus Torvalds nació en Helsinki, Finlandia, el 29 de Julio de 1970. A principios de los noventa, Linus Torvalds era un estudiante de informática en la Universidad de Helsinki que practicaba con Minix. A pesar de su innegable talento, y casi con toda seguridad, este finlandés no podía imaginarse en esa época, que una década después seria conocido (y respetado) como una de las figuras clave en el mundo de la informática, ni que sus intervenciones en conferencias y otros actos públicos despertarían al máximo la expectación, pero así ha sido, debido a la serie de acontecimientos que al salir Torvalds estaba a punto de desencadenar a mediados del año 1991.

En esos días, Linus se encontraba trabajando en un proyecto propio para mejorar Minix (al que encontraba ciertas deficiencias) , o para desarrollar, de algún modo, algún sistema similar a Minix que funcionara mejor que Minix. Linus comenzó desarrollando ese proyecto en su 386 utilizando ensamblador pero pronto recurrió a C, lo que según el propio autor, aceleró el proceso. De esta forma, Linus Torvalds creó lo que pueda considerarse la primera versión de Linux (contracción de Linus y Unix), que posteriormente la misma autor numeraría como versión 0.0.1. Téngase en cuenta que a la primera versión de un programa que resulta completamente operativa y funcional se la considera, como convenio, version 1.0 de dicho programa .esto permite apreciar claramente que la versión 0.0.1 de Linux era poco menos que el juguete de un programador, y así lo entendió Linus. Efectivamente, esta versión primigenia solo contenía en realidad un kernel muy rudimentario y para poder realizar cualquier operación (de las escasas posibles) era imprescindible que la máquina también tuviera instalado Minix.

Sin embargo, la versión 0.0.1 de Linux no llegaría a difundirse. Linus Torvalds continuó modificándose el nuevo sistema operativo, y no lo hizo público hasta que consiguió ejecutar ciertos programas del proyecto GNU, como el *bash* (Bourne Again Shell) y el *gcc* (GNU C Compiler). Esta nueva versión de Linux sería numerada como versión 0.0.2, y fue la primera versión de Linux que un autor consideró mínimamente utilizable, a pesar de sus muchas limitaciones, y de suficiente nivel como para merecer ser publicada.

La publicación de la versión 0.0.2 de Linux tuvo lugar el 5 de Octubre de 1991, en un foro de debate sobre Minix (en concreto en *comp.os.minix*). Linus distribuyó de forma gratuita en este foro el código fuente de Linux, invitando a la participación activa en el proyecto Linux de cualquier persona con interés en el tema y por supuesto, con los suficientes conocimientos de programación como para examinar el código y aportar mejoras o añadidos al sistema, así como para apuntar posibles errores o nuevas ideas. Esta llamada a la colaboración tuvo un eco tal que, a partir de este punto podemos decir que Linux dejó de ser obra exclusiva de Linus Torvalds para pasar a ser el fruto del trabajo colectivo de un equipo formado por un buen número de colaboradores voluntarios.

2.4 TIPOS DE ATAQUES AL SISTEMA OPERATIVO DE RED

2.4.1 INGENIERÍA SOCIAL

▪ TIPOS DE INGENIERÍA SOCIAL

Son varios los métodos que un crackers puede utilizar para conseguir que le demos información o que le facilitemos un acceso que debería estar restringido. Lo más normal es que utilice una mezcla de todos ellos, con tal de crear el mejor entorno para poder alcanzar sus objetivos.

Los crackers, muy a menudo, consiguen información simplemente convenciendo a la víctima de que está en una posición en la que esa

información les es necesaria. Diciendo que son un superior, por ejemplo, un vicepresidente un poco lejano, podrían obtener cualquier información ya que es probable que un empleado no conozca a todos sus superiores.

▪ **SUPLANTACIÓN**

La suplantación se parece a la autoridad falsa en el sentido de que el cracker va a intentar convencer a alguien de que tiene los derechos suficientes para hacer lo que esta haciendo. La suplantación es una versión de la autoridad falsa en la que un cracker adopta la personalidad de un individuo que realmente existe.

Es difícil que un cracker pueda suplantar a alguien que conocemos si esta sentado justo delante de nosotros, pero es mucho mas fácil o difícil si esta hablando a través del teléfono, mediante un correo, en un chat o mediante un mensaje instantáneo. Un cracker podría ser capaz de imitar el estilo de escritura de la otra persona leyendo sus correos, y podría tener la suficiente información personal como para hacernos creer que se trata realmente de nuestro amigo o colaborador. Después de convencernos de que es quien dice ser, el cracker nos pedirá información o permiso para acceder a algo que no nos hará sospechar.

▪ **COMPASIÓN**

Uno de los métodos más infalibles que se suele utilizar un cracker consiste en hacer ver a la víctima que se necesita urgentemente lo que esta pidiendo, para hacerle sentir compasión y que se sienta obligado a ayudarlo. Por ejemplo, alguien que dice ser del departamento de publicidad llama diciendo que ha olvidado la contraseña y que necesita que sea cambiada urgentemente, porque tiene que presentar un informe a su jefe que le ha amenazado con matarlo si no lo hace a tiempo. Si el cracker da muchas vueltas al tema y consigue inflar lo suficiente la historia, el administrador, que sentía un poco de compasión, intentará hacer todo lo posible por ayudar, quizás incluso olvidando que es necesario verificar y estar totalmente seguro de que es el empleado que dice ser, y se saltará todas las normas establecidas para evitar este tipo de ataques.

Esta técnica de la compasión la utilizan millones de personas en sus vidas diarias, y los crackers la pueden utilizar igual de bien con tal de acceder a la información que necesiten.

▪ **SER PARANOICO**

Mucha gente es confiada de forma natural. Hay que darse cuenta de que no vivimos en un mundo utópico y hay que cultivar una sana paranoia y una cierta desconfianza en los demás. Los crackers no van a atacar a personas que desconfíen de él y busquen en seguida un objetivo más sencillo.

▪ **PREGÚNTELO TODO**

Solo porque alguien diga que necesita algo no significa que realmente lo necesite, o que tenga permiso para ello. Pregunte siempre por que necesita la información o el acceso que le esta solicitando, y sugiera otra s alternativa. Los crackers esperan que siga a ciegas todas sus indicaciones, y por muy bien que otras acciones pudieran colmar sus necesitan, ellos insistirán en que se haga lo que ellos digan. Intente encontrar la solución mas adecuada al problema que se le plantee, y si le ofrecen resistencia, extreme las precauciones. La mayoría de los ataques mediante ingeniería social se desmoronan haciendo preguntas insistentemente.

▪ **COMPRUEBE SIEMPRE LAS FUENTES**

A menos que sepa exactamente con quien esta hablando, ponga mucha preocupación en todo lo que diga y en todo lo que haga. Cuando alguien le haga una petición poco habitual por correo electrónico, llame a la persona en cuestión para que le confirme esa petición por teléfono. Si la persona le esta llamando por teléfono, pídale su numero y devuélvale la llamada para comprobar que es correcto. Cuando hable cara a cara con alguien que no conozca pídale algún tipo de identificación. Solicítele algún número de empleado o cualquier otro tipo de identificación interna q ue pueda validar o consultar con su jefe.

Incluso con estas precauciones, puede que el cracker haya hecho muy bien sus deberes y es posible que le de toda la información que le esta solicitando.

▪ **LOS CRACKERS HACEN SUS DEBERES**

Los crackers investigan mucho antes de lanzar cualquier ataque de ingeniería social. Intentarán obtener toda la información que puedan sobre la persona con la que van a interactuar en la empresa, sobre la estructura de la compañía y sobre la persona a la que pretendan suplantar. Mientras mas familiar les resulte el escenario con el que se van a encontrar, más sencillo les resultará engañar a la gente. Herramientas muy comunes a la hora de obtener información con las siguientes:

▪ **PRUEBAS DE SEGURIDAD**

Hay ocasiones en las que un crackers ha enviado algún programa que supuestamente ayudaba a comprobar si una maquina era vulnerable al ultimo gran agujero de seguridad descubierto. En realidad, lo que hacia ese programa era provocar otro agujero de seguridad. Lo más normal es que el cracker dijera que el programa se tenía que ejecutar como root para que, en efecto, detectara la vulnerabilidad, con lo que se facilitaba mucho más aun el trabajo al cracker.

2.4.2 PROGRAMAS Y CABALLOS DE TROYA

La inserción de un programa caballo de Troya es el método más sencillo que tiene un cracker para ejecutar códigos maliciosos en nuestra máquina. El programa será, generalmente, algo que atrae al usuario, como un juego, un protector de pantalla, un programa de mensajes instantáneos o un reproductor MP3. Pero, al contrario de los normales, este tipo de programas incluirá alguna función que creará o explotará algún agujero de seguridad del sistema.

Siempre que ejecute un programa en Linux para acceder a todo lo que usted tenga acceso, es decir, cualquier troyano podrá leer y escribir en sus ficheros, crear conexiones de red, enviar correos, intentar infiltrarse en otras máquinas y ejecutar cualquier comando de forma aleatoria. Un troyano que se ejecute como root podría tener control total sobre la máquina.

Un ejemplo de esto último es el troyano *Remote Shell*, también conocido como RST.b. si un ataque consigue engañar a un usuario para que se descargue y ejecute el programa RST.b, infectar, si puede, sus ficheros de sistema del directorio /bin y se podrá escuchar en el puerto 5503. El atacante podrá conectarse a este puerto y lanzar comandos UNIX que serán ejecutados con la identidad del usuario que ejecuto este troyano RST.b.

➤ MEDIDAS DE PRECAUCIÓN CONTRA LOS CABALLOS DE TROYA

Nunca ejecute un programa que provenga de fuentes poco fiables, la naturaleza anónima de internet hace difícil estar seguros de la identidad de las personas. Es muy sencillo enviar correos y que parezcan que procedan de cualquier dirección. Siempre que reciba algún programa debe verificar que proceda realmente de la persona que dice que se lo envía.

Averigüe que es antes de ejecutarlo. Si ha conseguido un programa que procede de una fuente de confianza, debe asegurarse de que se trata antes de ejecutarlo. Solo porque el origen sea fiable no implica necesariamente que el programa también lo sea. /bin/rm puede llegar a ser un programa mortal, incluso aunque sea un programa válido.

Ejecútelo antes en un servidor virtual. Puede utilizar productos del estilo de User Mode Linux disponible para núcleos 2.2 o posteriores e integrado directamente en el núcleo 2.5; o de VMWare con el fin de crear máquina virtuales. Puede ejecutar el programa sospechoso en la máquina virtual para depurarlo y comprobar si hace algo sospechoso antes de ejecutarlo en el entorno de producción.

Ejecútelo antes en un entorno chroot cerrado. Ejecute el programa como un usuario con pocos privilegios en un entorno cerrado con chroot. Intente averiguar que esta haciendo exactamente antes de decidir ejecutarlo de forma normal.

Nunca ejecute nada como root. Al ejecutar un programa extraño como root le estamos permitiendo hacer absolutamente cualquier cosa: aplicar parches

al núcleo, crear nuevos usuarios, instalar nuevos programas. El usuario root debería ser utilizado solo en aquellos casos en los que sea estrictamente necesario.

En caso de duda, deseche el programa. Es más que probable que la mayoría de los programas que reciba por vías dudosas estén ya disponible en paquetes preparados para la instalación en el sitio web de su distribución de Linux. Si recibe algo que no está todavía empaquetado, solicite el código fuente y compílelo usted mismo.

➤ **NOTICIAS FALSAS DE ACTUALIZACIONES**

Muchas distribuciones de Linux y proyectos de software tienen listas de correos a las que podemos suscribirnos para recibir notificaciones cuando un determinado problema de seguridad ya tenga solución, o cuando sea necesaria alguna actualización por cualquier razón. Para conseguir que nuestros sistemas estén totalmente actualizados es necesario que lea estos mensajes y actúe en consecuencia.

➤ **VERIFIQUE LAS NOTICIAS QUE RECIBE**

Asegúrese de que conoce y de ser capaz de reconocer el formato que utiliza su distribución de Linux para emitir sus anuncios, esto reduce el riesgo de que pueda ser engañado con un mensaje falso. Además, tendrá que comprobar a lo con sumo cuidado el nombre de ficheros o de los sitios Web a los que hace referencia el mensaje en cuestión. Por ejemplo si tiene

instalada en su sistema la distribución Engarde Linux y una notificación de actualización le indica que se descargue el fichero <ftp://ftp.engarde-linux.org/some/file.tgz> debería sospechar de este mensaje , ya que el verdadero sitio FTP de Engarde Linux no contiene guión bajo en su nombre.

Tenga presente que además que la mayoría de los distribuidores y vendedores firman sus correos electrónicamente con PGP (*Pretty Good Privacy*). Debería visitar la página de seguridad de su vendedor y descargue e instalar su clave, de forma que pueda verificar cualquier mensaje que le envíe. Si utiliza el programa compatible con PGP, GnuPG (*GNU privacy Guard*), puede instalar las claves muy fácilmente. En el siguiente proceso descargaremos la clave de Guardian Digital, los creadores de Engarde Linux:

```
$ wget http://ftp.engradelinux.org/pub/engrade/ENGARDE-GPG-KEY
```

```
$ gpg ENGARDE-GPG-KEY
```

```
gpg: pub 1024D/DE7b7EED 2001-01-25 Guardian Digital, Inc
```

```
<security@guardiandigital.com>
```

```
$ gpg- import ENGARDE-GPG-KEY
```

```
gpg: key DE7b7EED: public key imported
```

```
gpg: Total number processed: 1
```

```
gpg: imported: 1
```

```
$ gpg --edit "Guardian Digital"
```

```
pub 1024DDE7B7EED created: 2001-01-25 expires: never trust: -/q
```

sub 1024g/A6FBE2F0 created: 2001-01-25 expires: never

(1). Guardian Digital, Inc. <security@guardiandigital.com>

Commad> **fpr**

pub 1024D/DE7B7EED 2001-01-25 Guardian Digital, Inc.

security@guardiandigital.com

Fingerprint; F008 BF42 4B30 1CD7 2F56 C11A 1C3E 5CA9

DE7B 7EED

Command> **lsign**

Are you really sure that want to sign this key with your key?

The signature will be marked as non-exportable.

Really sign? **Yes**

Passphrase: <***passphrase***>

Command> quit

Con este proceso estamos firmando la clave, indicándole a GPG que es de nuestra confianza. Antes de hacer esto hay que verificar que la clave que estamos firmando es realmente la clave del vendedor. Esto se hace asegurándonos de que el ID de la clave (1024 D/DE7BEED) y su huella (*fingerprint*: F008 BF42...) son correctos. Pero aquí se plantea un problema parecido al del huevo la gallina. El ID de la clave y la huella de esa clave, incluso en aquellas claves proporcionadas por un cracker con ideas malvadas, siempre coincidirán. Por tanto, será necesario algún otro método para comparar el ID y la huella de la clave proporcionada por el vendedor.

Por ejemplo, la clave podría incluirse en el propio medio en el que se haya distribuido el software, en la documentación, o en alguna página Web oficial, o en algún correo que se haya recibido de la compañía y que hayamos podido verificar.

Cada vez que reciba una notificación sobre alguna actualización de un distribuidor de software, debería verificar la firma PGP del mensaje. Los mensajes se firman habitualmente de dos formas: o en el propio cuerpo del mensaje o con una firma separada como un anexo del mensaje.

➤ OTROS TROYANOS

Otros ejemplos de ataques podrían considerarse troyanos. La mayoría de ellos necesitan algún tipo de ingeniería social para conseguir que caigamos en la trampa.

2.4.3 ARCHIVO TAR MALIGNOS

Un archivo tar es simplemente un grupo de ficheros guardados en uno solo, llamado frecuentemente *tarball* (paquete tar). El nombre de un paquete tar finaliza normalmente con la extensión .tar. Hay ocasiones en las que se comprimen con gzip y tendrán en ese caso la extensión .tar.gz o .tgz. Para extraer un *paquete tar* hay que ejecutar el programa tar con la opción “x” como, por ejemplo tar xvf tarball.tar o tar xzvf tarball en el caso de estar comprimido. Los paquetes tar contienen información sobre propietario, fechas y permisos de los ficheros que incluye, y pueden contener directorios

e incluso enlaces simbólicos. De esta forma, es posible incluir en un paquete tar un árbol de directorios completo, haciéndonos más fácil trasportar una estructura compleja de ficheros de una maquina a otra.

El programa tar tiene cuidado de eliminar la barra de inicial del nombre de los ficheros a no ser que se utilice la opción `-p`. de esta forma se consigue que todos los ficheros incluidos en el tarball tengan nombres de ruta relativos y que no se puedan sobrescribir accidentalmente ficheros del directorio raíz, a no ser que desempaquetemos el fichero tar justo en este directorio.

Aunque el propio programa tar no permite crear paquetes que contengan ficheros con nombres potencialmente peligrosos, cualquier cracker podría utilizar un editor hexadecimal y coger un paquete tar valido y renombrar los ficheros manualmente. Por ejemplo, aquí tenemos un paquete tar que contiene varios ficheros sospechosos:

```
Host # cd /var/tmp/
```

```
Host # mkdir unpack
```

```
Host # tar tvzf malicious.tgz
```

```
-rw- - - - root/bin 4574 2002-10-13 18:28:28 /etc/nologin
```

```
-rwxr-xr-x root/bin 829392 2002-10-13 18:28:28 ../ ../ ../ ../bin/grep
```

```
-rw- - - - root/bin 29381 2002-10-13 18:28:28 passwd - /etc/passwd
```

```
-rw- - - - root/bin 4020 2002-10-13 18:28:28 passwd
```

```
-rwsr-xr-x root/bin 48223 2002-10-13 18:28:28 rootshell
```

Este paquete tar contiene varios archivos diferentes en su interior. Para empezar, contiene un fichero `/etc/nologin` que comienza con una barra y que sería desempaquetado en el directorio real `/etc` y no en `/var/tmp/unpack`. El fichero `grep` del paquete comienza con muchas referencias al directorio padre (`..`), con la intención de subir en el árbol de directorios hasta llegar al raíz e instalar luego una nueva versión de `grep` en el directorio `/bin`. Y con la misma técnica se pretende sustituir el fichero `/s`. Luego vemos que el paquete contiene un fichero llamado `passwd`, que es enlace simbólico. Sin embargo, el cracker ha incluido un segundo fichero llamado `passwd`, que se desempaquetaría sobre el anterior. Pero como esta era un enlace simbólico, realmente se instalara sobre el fichero `/etc/passwd`. Por ultimo, tenemos un fichero llamado `rootshell` que se graba con el bit `suid` de root activado. Si root extrajera este paquete, el programa `rootshell` seguiría manteniendo ese bit activado, lo que presumiblemente podría ser aprovechado por el atacante para conseguir, instantáneamente, acceso como root.

- **UTILICE TAR DE FORMA SEGURA**

Las versiones mas antiguas del programa tar de Gnu no hacían nada para evitar incluir la barra y los `..` en los nombres de los ficheros. Las versiones posteriores corrigieron el problema, para volver a introducirlo de nuevo en versiones más modernas.

GNU tar es inmune a ataques de enlaces simbólicos, ya que borrara cualquier fichero existente antes de instalar el que se incluye en el paquete tar.

El ultimo ataque es valido para las versiones actuales de GNU tar, y muy probablemente no será corregido, ya que se considera un funcionamiento correcto. Cuando root extrae un paquete tar, el comportamiento predeterminado preservara tanto los propietarios del fichero como sus permisos, lo que significa que se acabara creando un fichero suid de root cuando extraiga el paquete anterior. La mejor forma de evitar este problema pasa por no extraer nunca los paquetes tar como root. Los usuarios normales mantendrán los permisos solo si utilizan la opción p de tar. Si tiene que desempaquetar un fichero tar como root (para preservar los propietarios de los ficheros, por ejemplo) asegúrese de comprobar posteriormente los ficheros suid y sgid.

2.4.4 PAQUETES RPM CORRUPTOS

Los paquetes RPM son los predeterminados en muchas distribuciones Linux, como Red Hat y Mandrake. Desde la versión 4.0 hasta 4.0.2 -7x el programa rpm presentaba un error que podía provocar un desbordamiento de la pila si se la hacia una consulta a un paquete RPM malformado. Si construía de una determinada forma el fichero RPM, un cracker podía conseguir que el programa rpm ejecutara el código que quisiera. Esta vulnerabilidad no

requería que el programa RPM estuviera instalado, solo era necesario hacerle una consulta: por ejemplo, con el comando `rpm -qpi fichero.rpm`.

Un cracker podría dejar ficheros o programas maliciosos “perdidos” por algún lugar, con la esperanza de que usuario root los vea y quiera averiguar que son. Si el administrador, al encontrar esos paquetes, quisiera comprobar que contienen esos ficheros RPM (cosas que en principio no debería presentar ningún problema para la seguridad del sistema, ya que no iba a instalar los paquetes), podría acabar provocando un compromiso total de su cuenta de usuario. Vemos que un cracker solo necesita convencer al administrador de que utilice el programa rpm mediante ingeniería social.

Ora técnica, consistiría en que alguien “viese” el RPM utilizando less. Less, es un paginador, al igual que more. Sin embargo, less utiliza un programa, lesspipe, que puede ser utilizado para preprocesar ficheros. Por ejemplo, las paginas de manual terminaban con extensiones como 1, 2, o man. Si se configura less para que use el preprocesador lesspipe, ejecutará el fichero mediante los comandos troff apropiados para hacer legible la pagina de manual, en vez de mostrar el propio código troff. El resultado se devuelve por un *pipe* al paginador y se visualiza. Como lesspipe ejecuta automáticamente el fichero rpm utilizando el comando `rpm -qpi`, el administrador solo tendrá que ejecutar less sobre el fichero para provocar que el comando rpm sea invocado y la cuenta de usuario se vea comprometida.

Un ataque parecido a este podría ser invocado directamente por el cracker. Algunos servidores de impresión reconocen ciertos tipos de fichero y son capaces de modificarlos para imprimirlos correctamente. Como imprimir un paquete RPM no tiene mucho sentido, el software de impresión ejecutara el comando `rpm -qpi` para generar un listado de ficheros, similar al modo de operar del programa `lesspipe`. Esto significa que el cracker podrá acceder a la copia de impresión de usuario, normalmente `lp`, y usarla para lanzar un violento ataque.

o CUIDADO CON LO DESCONOCIDO

La solución a este error determinado es simple: actualice su programa RMP. Sin embargo, esta vulnerabilidad nos demuestra que algo que en principio pudiera parecer tan inofensivo, como puede ser listar ficheros de un paquete o ver un fichero a través de un paginador, puede disparar efectos colaterales no deseados. Es muy importante que siempre que trabaje con ficheros de origen desconocido lo haga como un usuario normal, y a ser posible en una máquina virtual o de pruebas.

o ROBO DE CUENTA DE USUARIO

Siempre que tenga una máquina que utiliza muchos usuarios, existe el riesgo de que un usuario intente atacar las cuentas de otros usuarios. Esto es bastante habitual, por ejemplo, en los laboratorios. Las siguientes técnicas fueron utilizadas por uno de los autores (que prefiere mantenerse en el

anonimato) de hacker en Linux, para conseguir acceder a las cuentas de muchos de sus compañeros.

o **PANTALLA DE CONEXIÓN FALSA**

Uno de los trucos mas sencillos consiste en conectarse a un sistema, ejecutar un programa que imite la pantalla de conexión al sistema y dejar encendido el ordenador. El siguiente usuario que llegue a ese ordenador tecleara su nombre de usuario y su contraseña, que serán guardados, aparecerá ante él un mensaje de error indicándole que la contraseña tecleada ha sido incorrecta y saldrá devolviéndole al usuario la verdadera pantalla de conexión. Poco a poco, el cracker ira haciéndose con nombres de usuarios y contraseñas que podrá utilizar cuando quiera.

o **SHELL SUID**

Si alguien esta conectado a un sistema, abandona un momento su sitio para ir al baño y olvida bloquear su pantalla, al cracker le bastaría con sentarse en el sitio abandonado y realizar ciertas acciones como es el usuario. Para poder usar su cuenta mas adelante, el cracker podría crear un programa propiedad de ese usuario con el bit suid activado. Si el cracker lo tuviera bien montado, solo necesitaría un directorio en el que pudiera escribir todo el mundo, como */ home / cracker / shells /* con el siguiente contenido:

/ home / cracker / shells\$ cat shell.c

```
# include <stdio.h>
```

```
Main ( ) {
```

```
    Setuid (getuid ( ) ) ;
```

```
    System ("/bin/sh") ;
```

```
}
```

```
/ home / cracker / shell$ gcc -o shell shell.c
```

```
/ home / cracker / shell$ cat copyshell
```

```
# ! / bin / sh
```

```
Myusername= `whoami`
```

```
Cp / home / cracker /shells / shell / home/ cracker/ shells/ $myusername
```

```
Chmod 4755 / home / cracker / shells / $myusername
```

Cuando alguien deje su maquina desatendida, el cracker solo tendrá que

sentarse y ejecutar / home / cracker / shells/ copyshell, y el nuevo programa

suid será creado. Mas adelante, el cracker podría asumir el papel del otro

usuario ejecutando el fichero del directorio shells:

```
/ home / cracker / shells$ whoami
```

```
Cracker
```

```
/ home / cracker / shells$ jdoe
```

```
/ home / cracker / shells$ whoami
```

```
Jdoe
```

o SALVA PANTALLA TROYONIZADO

Un ejemplo anterior proporcionaba un acceso prolongado a la cuenta del

usuario atacado, pero no conseguía hacerse con la contraseña de ese

usuario. Otra técnica para que un usuario lo suficientemente despistado , deje

abandonada su maquina consiste en hacerle teclear su contraseña cuando

regrese a su lugar de trabajo. Una forma sencilla de conseguir esto es compilar una versión de un salvapantallas, como pueda ser *xlock*, que grabe las contraseñas en un fichero. Cuando el usuario se despiste, el cracker se ira corriendo a su sitio y ejecutará su nueva version del salvapantallas.

2.4.5 VIRUS Y GUSANOS

Además de con los troyanos, existen otros dos tipos de programas con los que debe tener cuidado. Se trata de los virus y gusanos.

Los virus son parecidos a los troyanos en el sentido de que hacen a lgo en nuestro sistema que no queremos que hagan, y sin nuestro consentimiento ni autorización. Un virus, una vez que ha activado, infectará otros programas o ficheros de nuestro ordenador, mientras que un troyano es un programa independiente que no puede propagarse por si solo. Ni los virus ni los troyanos pueden infectar una maquina desde el exterior sin algún tipo de inversión humana.

Un gusano es un programa que puede infectar tanto la maquina local como sistemas remotos. Normalmente se propaga de maquina en maquina a través de la red, ya sea atacando o utilizando otros programas de red, o utilizando las características de compartición de ficheros del sistema. En otras palabras, un gusano se reproduce automáticamente, mientras que con un troyano hay que engañar el usuario para que lo descargue y finalmente ejecute el programa malicioso. Por lo tanto, los gusanos pueden llegar a ser mucho más destructivos, ya que no se basan en la candidez de los usuarios.

Sin embargo, los programas mas malignos existentes hoy en día, son realmente híbridos de los tres tipos: troyanos, virus y gusanos. Por ejemplo, el famoso virus Melissa era un troyano (con la apariencia de un correo amable, nos invitaba a abrirlo), y un virus (infectaba todos los ficheros de procesamiento de texto que hubiese en nuestro ordenador), y un gusano (utilizaba un fallo de seguridad del programa Microsoft Outlook para propagarse a todas las personas que aparecieran en nuestra libreta de direcciones). La gente del sector ha comenzado a agrupar a los virus y a los gusanos en una sola categoría, a la que denominan genéricamente *virus*, pero siendo precisos deberían denominarlo híbridos virus/gusanos.

Como pueden propagarse de programa en programa, los virus y a los gusanos tienen mucha mas capacidad de hacer daño que los simples troyanos.

- **COMO SE PROPAGAN LOS VIRUS Y GUSANOS**

Los gusanos bien diseñados se propagan muy rápidamente y consiguen infectar un gran numero de maquinas en muy poco tiempo, normalmente mas rápidamente que lo que tardaría los fabricantes de antivirus en responder. Pueden reproducirse utilizando multitud de mecanismos. Estos son los métodos más populares:

Ficheros infectados. Un virus puede infectar otros ficheros, por ejemplo, sus documentos de procesamiento de texto, y luego infectar los ficheros de

otros usuarios cuando reciban por correo uno de esos documentos infectados.

Ficheros compartidos. Un gusano puede aprovecharse de los servidores de ficheros existentes e infectar los archivos que estén ubicados en estos servidores. Cuando los usuarios abran estos ficheros también se infectarán.

Disquetes. Los disquetes con virus infectaran todos los ordenadores en los que sean insertados, por ejemplo, al llevar un disquete desde su trabajo o colegio al ordenador de su casa y USB.

Correo electrónico. Un virus puede aprovecharse de algún fallo de su programa de correo y enviarse a si mismo a gente con la que haya mantenido correspondencia últimamente, o puede leer sus ficheros de alias para obtener direcciones de correo, por ejemplo. Como el correo va a aparecer como enviado por una persona conocida, es más que probable que a nueva víctima abra el correo y ejecute los programas anexados, con lo que se convertirá en un nuevo sistema infectado. Este es el método mas extendido para la propagación de virus.

2.5 ATAQUES DESDE LA RED

2.5.1 RUTAS DE CONFIANZA Y PROGRAMAS TROYANOS

Una carta bajo la manga de todos los cracker son los caballos de Troya. Este tipo de programa sustituye en un ejecutable por otro, y realiza las acciones que se le soliciten sin más que seguir una serie de pasos establecidos. Los programas troyanos forman parte de los que se conocen como: rootkits, en lo que los programas sustituidos son realmente ejecutables del propio sistema. Estos programas también pueden ejecutarse aprovechando a usuarios confiados, sin necesidad de tener que sustituir ningún ejecutable del sistema.

USUARIOS CON <<.>> EN SU VARIABLE PATH.

La mayoría de usuarios les gusta tener el directorio <<.>> en su variable path.

Por desgracia esto también es cierto para la mayoría de los administradores, incluso cuando se han conectado al sistema como root. De esta forma cada vez que quieren ejecutar un script de shell o algún otro programa, se ahorran un par de pulsaciones de teclas, ya que no tienen que teclear `sh programa` o `programa`, les vale sólo con poner `programa`.

o MEDIDAS DE PROTECCIÓN CONTRA LAS LLAMADAS AL SISTEMA INSEGURAS.

Los programas que vayan a ser aplicaciones `suid` o `Sgid`, además de las aplicaciones externas en las que se apoyan, no deberían ejecutar programas sin indicar la ruta completa a no ser que antes se deshagan de los privilegios

que tienen. Además, antes de invocar al programa externo, las aplicaciones deberían controlar la variable PATH, ya sea asignándole algún valor que sea seguro o eliminando de ella los directorios que puedan presentar algún problema. Siempre que sea posible; hay que evitar que las aplicaciones Xinetd invoquen a programas externos. En vez de esto, la funcionalidad de la aplicación sXid deberá restringirse a una pequeña aplicación o que un demonio no ejecute ningún comando externo.

Los administradores de sistemas y los no-programadores poco pueden hacer, de parte de borrar de su sistema todas estas aplicaciones, o de sustituirlas por script que puedan controlar la variable PATH. Afortunadamente, estos tipos de vulnerabilidades no son fáciles de descubrir por un atacante, y son relativamente raras. El riesgo se puede minimizar simplemente disminuyendo el número de aplicaciones suid y Sgid del sistema.

2.5.2 SUDO

Sudo (y otra herramienta parecida, súper) es una utilidad muy conocida que sirve para delegar responsabilidades administrativas en otro usuario. Utilizando Sudo es posible conceder a determinados usuarios los derechos necesarios para que realicen ciertas tareas administrativas que normalmente requieran el acceso como: root. Por ejemplo, se podrá conceder a determinados usuarios los derechos necesarios para que puedan añadir, eliminar o modificar usuarios o cambiar su contraseña.

Sin embargo, el uso de Sudo puede verse afectado por múltiples abusos, con el resultado final de que el usuario obtendrá mas derechos de los que le habíamos asignado en su principio. Si se utiliza sudo sin tener en cuenta todas las características del programa que se le esta p ermitiendo ejecutar, es posible que el usuario acabe consiguiendo un acceso como root sin ninguna restricción.

▪ **ATAQUES AL CAMBIO DE CONTRASEÑA CON SUDO**

A menudo, para aliviar un poco la carga de trabajo del administrador del sistema se permite a algunos usuarios normales ejecutar el comando *passwd* como *root* para que puedan modificar las contraseñas de otros usuarios.

Una vez que se concede este acceso, al usuario en el que se confía va a poder cambiar cualquier contraseña, incluso la del propio *root*. Esto puede presentar un serio problema si el usuario resulta no ser tan de fiar como uno pensó en un principio. Además, también puede ocurrir que un cracker ya haya atacado la cuenta de ese usuario. Si ocurre esto, Linux no va a poder detectar diferencias entre las conexiones del usuario auténtico y las del cracker, ya que se hacen con el mismo nombre de usuario.

▪ **MEDIDAS DE PROTECCIÓN CONTRA EL CAMBIO DE CONTRASEÑAS CON SUDO.**

Puede crear un script que actúe como interfaz, y que ese script compruebe que la contraseña que se va a modificar es la de un usuario normal, y no la

de un usuario de sistema. Las cuentas de sistema tienen normalmente números de ID menores que un cierto valor preestablecido, casi siempre 200 o 500, dependiendo del sistema. Si se intenta cambiar la contraseña de cualquier cuenta que tenga un ID de usuario menor que el valor anterior debería generarse un error.

Dependiendo de las políticas establecidas en el sistema, este script podría también verificar que el usuario cuya contraseña se quiere modificar no tiene ningún tipo de bloqueo y que cuenta con una shell válida.

A menudo se permite a los usuarios que ejecuten programas de configuración que incluyen la posibilidad de ejecutar un editor, como *crontab -e -u usuario*. Como la mayoría de estos programas permiten que el editor que se quiere utilizar pueda especificarse mediante las variables VISUAL o EDITOR podría lanzarse la ejecución de cualquier programa.

El editor debería restringirse a algunos bien conocidos, como *vi*, *ed* o *emacs*. Sin embargo, la mayoría de los editores ofrecen la posibilidad de ejecutar comandos externos o de ofrecer salidas a la shell. Como el editor se está ejecutando bajo la cuenta de *root*, se podría lanzar cualquier programa como *root* sin más que invocarlo desde la shell del editor.

Incluso aunque el editor esté protegido para que no permita la ejecución de comandos externos o para que no ofrezca salidas a la shell, todavía va a poder leer ficheros en los que no se pensó en un principio. Si un editor se ejecuta como *root*, va a poder leer ficheros como el */etc/shadow*, cuyos

permisos los tiene establecidos para que sólo pueda ser leído por el usuario *root* y no por un usuario normal al que se le han delegado ciertas tareas administrativas.

Pero puede ser peor aún, ya que la mayoría de los editores permiten la edición de un nuevo fichero en cualquier momento. Si un comando Sudo permite a un cracker editar el fichero */etc/hosts*, no va a existir ningún motivo que le impida, tras realizar las modificaciones necesarias en ese fichero, abrir después el fichero */etc/shadow* y escribir en él. Un pequeño cambio en la contraseña cifrada del usuario *root* va a permitir al cracker conectarse al sistema como *root* o ejecutar el programa *su* y convertirse en *root*.

▪ **MEDIDAS DE PROTECCIÓN CONTRA LA EJECUCIÓN DE EDITORES**

La mejor solución al problema de los editores dependerá de la aplicación. En el caso en que sea necesaria la edición de algún fichero, ese fichero debería ser bloqueado y copiado a una ubicación segura, donde pueda ser modificado por un usuario normal sin riesgo alguno. De esta forma, el usuario podrá editar esta copia temporal sin riesgo de comprometer otros ficheros restringidos del sistema. Una vez que se hayan realizado las modificaciones necesarias, habría que comprobar que no se han modificado campos restringidos de ese fichero, y que esos cambios son consistentes con la estructura del fichero y con los requerimientos del sistema. Una vez

realizadas estas comprobaciones se podría volver a copiar a su ubicación original y eliminar el bloqueo.

Si queremos extremar las precauciones, podríamos copiar el fichero que se tiene que editar a un lugar seguro, bajo una cuenta de usuario segura y ejecutar después el editor en un entorno cerrado con *chroot* (con el sistema de ficheros raíz cambiado). De esta forma conseguimos que el usuario tenga un doble bloqueo contra el resto del sistema, el que proporcionan los privilegios del usuario restringido y el entorno *chroot*. Sin embargo estas medidas suscitan una cuestión: si no se confía en el usuario hasta el extremo de tener que tomar estas medidas, ¿por qué delegar en esta persona responsabilidades administrativas?

▪ PROGRAMAS EXPLOTABLES LOCALMENTE

Los problemas con la seguridad local y con la escalada de privilegios se acentúan aún más por el hecho de que el atacante ya tiene cierto grado de acceso al sistema. Mientras que son sólo un puñado las técnicas que pueden utilizarse para acceder a un ordenador remotamente, en este caso son miles los programas que están accesibles cuando se ha conseguido acceder al shell.

Una vez dentro del sistema, son dos los tipos de programas que pueden ser atacados: todos aquellos instalados en el sistema, y todos los programas y

scripts creados por el administrador o por los usuarios locales. A lo largo de la historia, los programas de estas dos categorías han venido sufriendo los mismos tipos de problemas: desbordamientos de búfer, condiciones de carrera y una validación incorrecta de las entradas. Es más probable que el atacante sea capaz de determinar rápidamente qué programas del sistema presentan vulnerabilidades conocidas cotejando las versiones de los paquetes instalando contra alguna base de datos o en algún archivo de listas de correo. Cualquier programa hecho por nosotros que tengamos instalado le llevará al cracker mucho más tiempo averiguar si contiene alguna vulnerabilidad. Sin embargo, es mucho más probable que esas aplicaciones personalizadas presenten errores que los programas del sistema, y por una sencilla razón: las aplicaciones del sistema han sido revisadas y probadas por miles de ojos. Sin embargo los scripts del administrador local no han visto nunca la luz del día.

2.5.3. PROGRAMAS SXID.

Los programas `suid` y `sgid` son una fuente constante de problemas. Si no están programados con mucho cuidado, un desbordamiento, ó un error o la ejecución de un comando externo, podrían provocar rápidamente un cambio en la identificación del usuario y el correspondiente cambio de privilegios. Esto puede ocasionar a veces el inmediato compromiso de la cuenta a root, aunque otras veces nos deja a un solo paso.

Afortunadamente, y en contra del comportamiento de otras versiones de UNIX. Linux no admite los scripts de shell suid. Y es que sencillamente no hay forma de cerrar todos los agujeros o rendijas de seguridad que se producen al crear scripts suid. En Linux, un programa sXid tiene que ser un ejecutable compilado, Aunque Perl proporciona un intérprete especial, suidperl, para procesar los scripts suid de Perl. Lamentablemente, esto complica las medidas de protección contra los programas suid dañinos.

2.5.4 ATAQUES POR DESBORDAMIENTOS DE BÚFER CONTRA EJECUTABLES SUID DE ROOT: SUDO.

Sudo es un programa que permite que usuarios normales puedan ejecutar programas como root. Ya hemos hablado de sudo en este mismo capítulo, y especialmente del cuidado que hay que poner cuando se generen las reglas de sudo.

Desde la versión 1.5.7 hasta la 1.6.5p2 sudo tiene un desbordamiento de búfer que puede ser explotado. y como obviamente sudo tiene que tener el bit suid de root activado para que funcione correctamente, el exploit provoca un acceso como root. El error estaba en el código que interpretaba el valor de unas variables, una característica aparentemente insignificante. Cuando se utiliza la opción -p. sudo sustituirá %h por el nombre de la máquina y %u por el nombre de usuario. Esta opción se utiliza habitualmente en aquellos scripts creados por el propio usuario root para facilitar a operadores no -root, que entiendan lo que ocurre cuando se invoca a sudo.

Sudo reserva una cierta cantidad de memoria para el mensaje personalizado, ya que en tiempo de compilación no puede saber cuán largo es el nombre de máquina, el nombre de usuario o cualquier texto que introduzcamos. Desgraciadamente, las versiones vulnerables manejaban erróneamente las cadenas de las opciones %h y %u y no reservaban el espacio suficiente. Como podíamos incluir cualquier texto en el mensaje, era posible provocar un desbordamiento de pila y ejecutar comandos a nuestro antojo.

Una forma sencilla de comprobar si nuestra versión es vulnerable es utilizar el argumento -p h%h%. el analizador debería reservar el suficiente espacio para una h, seguida del nombre de máquina, y un símbolo % al final. Pero lo que ocurre en realidad es que trata a la primera h como %h, y nunca finaliza la cadena, dejando el % al final. Cuando se le pasa como argumento un mensaje formado adecuadamente, la memoria reservada es 5 bytes más pequeña de lo necesario y los datos del usuario acaban por sobrescribir el montón (heap).

➤ **MEDIDAS DE PROTECCIÓN CONTRA LOS DESBORDAMIENTOS DE BÚFER**

En este caso la memoria era insuficiente, y no se utilizaba una función segura de concatenación de cadenas, lo que hubiera evitado el desbordamiento de búfer. Una solución rápida para este problema con c y para otros ejecutables suid consiste en eliminar el bit suid:

Root # **chmod u - s / usr / bin / sudo**

Por supuesto que el código de Sudo fue corregido rápidamente, y las distintas distribuciones de Linux publicaron actualizaciones inmediatamente. Como siempre, actualizar sus programas es la solución más segura a largo plazo, ya que además no perdemos funcionalidad en ellos.

➤ **ATAQUES CONTRA APLICACIONES SUID NO -ROOT**

El administrador del sistema no tiene que preocuparse sólo por aquellas aplicaciones que, en mayor o menor medida, están relacionadas con la seguridad. Cualquier programa que sea ejecutable suid puede presentar un problema incluso aunque no sea propiedad del usuario root.

Consiguiendo comprometer un juego mal programado, se puede conseguir el nivel del usuario games. Luego, como el fichero es propiedad del usuario games, sería cuestión de sustituirlo con alguna nueva versión que genere alguna puerta trasera que se abra cuando algún usuario ejecute el juego. Como es bastante frecuente que los administradores estén aburridos y también quieran jugar alguna vez, esto puede suponer un compromiso inmediato de la cuenta root. Por lo tanto, lo único que tendrá que hacer el cracker es esperar que el administrador se ponga a jugar.

➤ **MEDIDAS DE PROTECCIÓN DE LAS APLICACIONES SUID NO-ROOT**

La mayoría de los programas que no son propiedad del usuario root pero que se ejecutan con el bit suid activado puede desarrollarse para que utilicen los

privilegios del usuario que los invoca. Si se asegura de que el ejecutable no tiene establecidos los permisos de escritura para el grupo o para el resto del mundo, se elimina la posibilidad de que sea sustituido en caso que se vea comprometido.

➤ **CREAR FICHEROS TEMPORALES DE FORMA SEGURA**

Muchos programadores intentan enviar estas condiciones de carrera creando un nombre de fichero único y no utilizado mediante las llamadas C al sistema `mktemp ( )` y `tmpnam ( )` :

```
Nombre_de_fichero_unico =mktemp("/tmp /mortal del xxxxxx " ) ;
```

```
Descriptor_de_fichero  open ( nombre_de_fichero_u nico, .... ) ;
```

De esta forma se garantiza que el nombre de ficheros será único cuando se genere, pero existe una condición de carrera en el tiempo que transcurre entre la generación del fichero y el momento en el que se invoca a la función `open ( )`.

2.5.5 ATAQUES AL NÚCLEO DEL SISTEMA

Los errores de los programas pueden ser desde tonterías sin importancia hasta problemas que afecten seriamente a la seguridad del sistema. Sin embargo, cuando se descubre un error relacionado con la seguridad del propio núcleo de Linux; hay razones para el pánico generalizado. Nuestra seguridad se basa en los controles que establece el núcleo para evitar que otros lean nuestros ficheros protegidos, que los usuarios normales no puedan realizar cambios en la memoria o para que las restricciones que tenemos establecidas funcionen a la perfección. Desgraciadamente, al igual

que cualquier otro programa, el núcleo de Linux también puede padecer de seguridad cuyos defectos pueden ser dramáticos

- **EXPLOITS CONTRA EL NÚCLEO**

El núcleo de Linux ha experimentado serios problemas de seguridad en el pasado. Alguno de estos errores puede ofrecernos un poco más de acceso al hardware; como los que se producen en los controladores de dispositivos mal programados. Otros pueden afectar la estabilidad del sistema, como un error que se produjo en el recién aparecido núcleo 2.4, que provocaba un fallo al desmontar un sistema de ficheros al reiniciar. Pero por desgracia, otras vulnerabilidades permiten que usuarios locales se conviertan en root.

La vulnerabilidad más reciente relacionada con la seguridad estaba provocada por la llamada ptrace (). Esta función, implementada por ejemplo en el programa strace, permite controlar las llamadas que se realizan al sistema cuando se ejecuta un programa; es extremadamente útil cuando estamos depurando código, especialmente de aquellos ejecutables de los que no tenemos su fuente.

Aunque monitorizar las llamadas al sistema es el uso más habitual que se le suele dar a ptrace, realmente es mucho más potente. Cualquier programa que pueda engancharse a otro mediante ptrace va a ser capaz de controlar la ejecución de ese programa y examinar su memoria y sus registros; debido al poder que ptrace nos ofrece, el núcleo no permite que se enganche un proceso dentro, que se ejecute como un usuario diferente. O para ser más

exacto, no debería permitirlo. Los núcleos de Linux anteriores al 2.2.19 y al 2.4.9 tenían un error en la forma de implementar ptrace. Normalmente, no se debería poder ejecutar ptrace sobre un proceso sXid; pero estos antiguos núcleos cometían un error al no establecer esta restricción.

Nengal desarrolló un exploit que ejecutaba copias de /usr/bin/passwd y /usr/bin/newgrp, en ambos casos procesos suid de root, de tal forma que el núcleo permitía enganchar ptrace al proceso newgrp, por que un proceso suid existente se estaba ejecutando bajo el mismo grupo de proceso. Una vez que enganchaba al proceso newgrp, el exploit conseguía una shell como root.

```
jdoe@machine$ 1s
Inserte_shellcode.c ptrace-exp.c
jdoe@machine$ make inserte_shellcode; make ptrace -exp
cc      insert_shellcode . c -o      insert_shellcode
cc      ptrace-exp . c      -o      ptrace-exp
jdoe@machine$ PATH=" $PATH:."
jdoe@machine$ ptrace-exp
attached
<wait>
# id
uid = 0 ( root ) gid = 0 ( root )
```

El programa ptrace – exp ha conseguido su propósito. Y todo lo que se necesita es el código y dos programas suid de root, aunque passwd y newgrp son los que utiliza este exploit, cualquier otro hubiera valido. El ataque se realiza contra el núcleo del sistema, no contra esos programas.

- **ACTUALIZAR EL NÚCLEO DE SU SISTEMA.**

La solución más segura para solucionar un error del núcleo de nuestro sistema pasa por actualizarlo. Pero esto es algo con lo que la mayoría de la gente no se siente muy a gusto.

- **DESACTIVAR EL CÓDIGO PELIGROSO DEL NÚCLEO**

En este caso el problema se podría explotar realizando la llamada ptrace al sistema, como esta llamada no se necesita en un funcionamiento normal del sistema; podría desactivarla y continuar en el núcleo actual sin ningún problema de seguridad, esto se puede hacer de dos formas:

- **ELIMINAR CAP_ SYS_ PTRACE CON LCAP**

La capacidad que permite la ejecución de ptrace sobre los procesos se llama: CAP_ SYS_ PTRACE. Si la deshabilita en el núcleo existente utilizando el programa 1cap, ya no estará disponible en el sistema para ningún usuario, con lo que el exploit fallará

Root # 1cap

Current capabilities : 0xFFFFFFFF

- | | |
|------------------------------|---------------------------|
| 0) * CAP_ CHOWN | 1) *CAP_ DAC_ OVERRIDE |
| 2) * CAP_ DAC_ READ_ SEARCH | 3) *CAP_ FOWNER |
| 4) * CAP_ FSETID | 5) *CAP_ KIL |
| 6) * CAP_ SETGID | 7) *CAP_ SETUID |
| 8) CAP_ SETPCAP | 9) *CAP_ LINUX_ IMMUTABLE |
| 10) *CAP_ NET_ BIND_ SERVICE | 11) *CAP_ NET_ BROADCAST |
| 12) *CAP_ NET_ ADMIN | 13) *CAP_ NET_ RAW |

- | | |
|-------------------------|---------------------|
| 14) *CAP_IPC_LOCK | 15) *CAP_IPC_OWNER |
| 16) *CAP_SYS_MODULE | 17) *CAP_SYS_RAWIO |
| 18) *CAP_SYS_CHROOT | 19) *CAP_SYS_PTRACE |
| 20) *CAP_SYS_PACCT | 21) *CAP_SYS_ADMIN |
| 22) *CAP_SYS_BOOT | 23) *CAP_SYS_NICE |
| 24) *CAP_SYS_RESOURCE | 25) *CAP_SYS_TIME |
| 26) *CAP_SYS_TTY_CONFIG | |

* = Capabilities currently allowed

Root # 1cap -v CAP_SYS_PTRACE

Current capabilities : 0Xffffeff

Removing capabilities:

19) CAP_SYS_PTRACE

Strace (2)

Root # 1cap

Current capabilities: 0XFFF7FEFF

- | | |
|---------------------------|-------------------------|
| 0) *CAP_CHOWN | 1) *CAP_DAC_OVERRIDE |
| 2) *CAP_DAC_READ_SEARCH | 3) *CAP_FOWNER |
| 4) *CAP_FSETID | 5) *CAP_KILL |
| 6) *CAP_SETGID | 7) *CAP_SETUID |
| 8) CAP_SETPCAP | 9) *CAP_LINUX_IMMUTABLE |
| 10) *CAP_NET_BIND_SERVICE | 11) *CAP_NET_BROADCAST |
| 12) *CAP_NET_ADMIN | 13) *CAP_NET_RAW |
| 14) *CAP_IPC_LOCK | 15) *CAP_IPC_OWNER |
| 16) *CAP_SYS_MODULE | 17) *CAP_SYS_RAWIO |
| 18) *CAP_SYS_CHROOT | 19) *CAP_SYS_PTRACE |
| 20) *CAP_SYS_PACCT | 21) *CAP_SYS_ADMIN |
| 22) *CAP_SYS_BOOT | 23) *CAP_SYS_NICE |
| 24) *CAP_SYS_RESOURCE | 25) *CAP_SYS_TIME |
| 26) *CAP_SYS_TTY_CONFIG | |

* = Capabilities currently allowed

Root # strace /bin/ls

Strace: exec: operation not permitted

Como puede comprobar, una vez eliminada la capacidad correspondiente, la llamada a la función ptrace ya no se encuentra disponible, incluso para el usuario root.

- **ELIMINAR CAP_SYS_PTRACE CON LIDS**

Si tiene un núcleo que admite configuración dinámica de las aplicaciones, como es el caso de LIDS, puede deshabilitar editando el fichero /etc/lids/lids.conf, cambiando la entrada correspondiente a CAP_SYS_PTRACE.

-19:CAP_SYS_PTRACE

Y luego haciendo que LIDS relea el fichero de configuración.

Root #lidsadm -s - +RELOAD_CONF

Este cambio se activará tras reiniciar el sistema, si quiere hacer el cambio temporalmente tendría que ejecutar lo siguiente:

Root \$ lidsadm -s -- -CAP_SYS_PTRACE

Pero ambas soluciones quieren que LIDS esté configurado para que admita cambios en tiempos de ejecución,

2.6. ALGUNAS MEDIDAS DE SEGURIDAD AL SERVIDOR EN LINUX RED HAT

2.6.1 IPTABLES (firewall)

Red Hat Linux contiene herramientas avanzadas para el filtrado de paquetes de redes — el proceso de controlar los paquetes de red cuando entran, se mueven y salen de la red dentro del kernel. Los kernels anteriores al 2.4 confiaban en ipchains para el filtrado de paquetes y usaban listas de reglas aplicadas a los paquetes en cada paso del proceso de filtrado. La introducción de kernel 2.4 kernel trajo consigo el iptables (también llamado netlter), lo cual es similar a ipchains pero expande enormemente el ámbito y el control disponible para el filtrado de paquetes de red.

- **FILTRADO DE PAQUETES**

El tráfico se mueve a través de una red en paquetes. Un paquete de red es una colección de datos en diferentes tamaños y formatos. Para enviar un fichero por red, el ordenador emisor debe en primer lugar partirlo en diferentes paquetes usando las reglas del protocolo de red. Cada uno de estos paquetes contiene una parte pequeña de los datos del fichero. Cuando recibe la transmisión, el ordenador receptor, reensambla los paquetes y construye de nuevo el fichero.

Cada paquete contiene información que le ayuda a navegar por la red y moverse hacia su destino. El paquete puede decirle a los ordenadores lo largo del camino, el ordenador destino, de dónde viene, a dónde va, qué tipo de paquete es, y otras muchas cosas más. La mayoría de los paquetes se

diseñan para transportar datos, pero algunos protocolos pueden usar los paquetes de forma especial. El protocolo Transmission Control Protocol (TCP), por ejemplo, utiliza un paquete SYN, que no contiene datos, para iniciar la comunicación entre dos sistemas.

El kernel de Linux contiene la característica interna de filtrado de paquetes, permitiendo aceptar algunos de ellos en el sistema, mientras que intercepta y detiene a otros. El filtro de red del kernel 2.4 tiene tres tablas o listas de reglas. Que son las siguientes:

filter — La tabla por defecto para el manejo de paquetes de red.

nat — Usada para alterar paquetes que crean una nueva conexión.

mangle — Usada por tipos específicos de alteración de paquetes.

Cada una de estas tablas tiene un grupo de cadenas internas que corresponden a las acciones llevadas a cabo por el filtro de red en el paquete.

Las cadenas internas para la tabla filtro son las siguientes:

INPUT — Aplica a los paquetes recibidos a través de una interfaz de red.

OUTPUT — Esta cadena sirve para paquetes enviados por medio de la misma interfaz de red que recibió los paquetes.

FORWARD— Esta cadena sirve para paquetes recibidos en una interfaz de red y enviados en otra.

Las cadenas internas para la tabla nat son las siguientes:

PREROUTING — Esta cadena altera paquetes recibidos por medio de una interfaz de red cuando llegan.

OUTPUT — Esta cadena altera paquetes generados localmente antes de que sean dirigidos por medio de una interfaz de red.

POSTROUTING — Esta cadena altera paquetes antes de que sean enviados por medio de una interfaz de red.

Las cadenas internas para la tabla mangle son las siguientes:

PREROUTING — Esta cadena altera paquetes recibidos por medio de una interfaz de red antes de que sean dirigidos.

OUTPUT — Esta cadena altera paquetes generados localmente antes de que sean dirigidos por medio de una interfaz de red.

Cada paquete de red recibido o enviado de un sistema Linux está sujeto al menos una tabla. Un paquete puede que sea verificado contra muchas reglas dentro de la lista de reglas antes de llegar al final de una cadena. La estructura y propósito de estas reglas puede variar, pero normalmente buscan identificar un paquete que viene de o se dirige a una dirección IP en particular o un conjunto de direcciones al usar un determinado protocolo y servicio de red.

Independientemente de su destino, cuando un paquete cumple una regla en particular en una de las tablas, se asignan a un objetivo (target) particular. Si la regla especifica un objetivo ACCEPT para un paquete que coincida, el paquete se salta el resto de las verificaciones de la regla y se permite que

continúe hacia su destino. Si una regla especifica un objetivo DROP, a ese paquete se le niega el acceso al sistema y no se envía nada de vuelta al servidor que envió el paquete. Si una regla especifica un objetivo QUEUE, el paquete se pasa al espacio del usuario. Si una regla especifica el objetivo opcional REJECT, el paquete es entregado, pero se envía un paquete de error al que envió el paquete.

Cada cadena tiene una política por defecto de ACCEPT, DROP, REJECT, o QUEUE. Si ninguna de estas reglas en la cadena se aplica al paquete, entonces el paquete es tratado de acuerdo a la política por defecto.

El comando iptables configura estas tablas, así como también configura nuevas tablas si es necesario.

- **OPCIONES USADAS EN COMANDOS IPTABLES**

Las reglas que permiten a los paquetes ser filtrados por el kernel se ponen en funcionamiento ejecutando el comando iptables. Cuando use el comando iptables, debe especificar las opciones siguientes:

Packet Type — Dicta qué tipo de paquetes filtra el comando.

Packet Source/Destination —Dicta qué paquetes filtra el comando basándose en el origen o destino del paquete.

Target — Indica qué acción es tomada en paquetes que cumplen los criterios mencionados anteriormente.

- **TABLAS**

Un aspecto muy potente de iptables es que se pueden utilizar múltiples tablas para decidir el destino de un paquete particular. Gracias a la naturaleza extensible de iptables, se pueden crear tablas especializadas y almacenarlas en el directorio `/lib/modules/< kernel -version> /kernel/net/ipv4/netfilter/`, donde, `<kernel-version>` corresponde al número de la versión del kernel.

La tabla por defecto, llamada `filter`, contiene las cadenas estándar por defecto para `INPUT`, `OUTPUT`, y `FORWARD`. Esto es similar a las cadenas estándar que se utilizan con `ipchains`. Sin embargo, por defecto, iptables también incluye dos tablas adicionales que realizan tareas de filtrado específico de paquetes. La tabla `nat` se puede utilizar para modificar las direcciones de origen y destino grabadas en un paquete, y la tabla `mangle` permite alterar los paquetes de forma especializada.

Cada tabla contiene las cadenas por defecto que realizan las tareas necesarias basadas en el propósito de la tabla, aún cuando se pueden añadir nuevas cadenas a cualquier tabla.

- **ESTRUCTURA**

Muchos comandos iptables tienen la siguiente estructura:

```
iptables [-t<table-name> ]  
<command>  
<chain-name>
```

```
<parameter-1> \  
    <option-1>  
<parameter-n>  
<option-n>
```

En este ejemplo, la opción <table-name> permite al usuario seleccionar una tabla diferente a la tabla predeterminada filter a usar con el comando. La opción

<command> indica una acción específica a realizar, tal como anexar o eliminar la regla especificada por la opción . <chain-name> Luego de la opción <chain-name> se encuentran un par de parámetros y opciones que definen qué pasará cuando un paquete coincide con la regla.

Cuando miramos la estructura de un comando iptables, es importante recordar que, al contrario que la mayoría de los comandos, la longitud y complejidad de un comando iptables puede cambiar en función de su propósito. Un comando simple para borrar una regla de una cadena puede ser muy corto, mientras que un comando diseñado para filtrar paquetes de una subred particular usando un conjunto de parámetros específicos y opciones puede ser mucho más largo. Al crear comandos iptables puede ser de ayuda reconocer que algunos parámetros y opciones pueden crear la necesidad de utilizar otros parámetros y opciones para especificar algo de los requisitos de la opción anterior. Para construir una regla válida, esto deberá continuar hasta que todos los parámetros y opciones que requieran otro conjunto de opciones hayan sido satisfechos.

Teclee iptables -h para ver una lista detallada de la estructura de los comandos iptables.

- **COMANDOS**

Los comandos le dicen a iptables que realice una tarea específica. Solamente un comando se permite por cada cadena de comandos iptables. Excepto el comando de ayuda, todos los comandos se escriben en mayúsculas. Los comandos de iptables son los siguientes:

A — Añade la regla iptables al final de la cadena especificada. Este es el comando utilizado para simplemente añadir una regla cuando el orden de las reglas en la cadena no importa.

C-Verifica una regla en particular antes de añadirla en la cadena especificada por el usuario. Este comando puede ser de ayuda para construir reglas iptables complejas pidiéndole que introduzca parámetros y opciones adicionales.

D — Borra una regla de una cadena en particular por número (como el 5 para la quinta regla de una cadena) . Puede también teclear la regla entera e iptables borrará la regla en la cadena que corresponda.

E — Renombra una cadena definida por el usuario. Esto no afecta la estructura de la tabla.

F—Libera la cadena seleccionada, que borra cada regla de la cadena. Si no se especifica ninguna cadena, este comando libera cada regla de cada cadena.

h — Proporciona una lista de estructuras de comandos, así como también un resumen rápido de parámetros de comandos y opciones.

I — Inserta una regla en una cadena en un punto especificado por un valor entero definido por el usuario. Sino se especifica ningún número, iptables colocará el comando en el tope de la cadena.

- **OPCIONES DE IDENTIFICACIÓN DE PAQUETES**

Diferentes protocolos de red proporcionan opciones especializadas las cuales se pueden configurar de formas específicas para coincidir un paquete particular usando ese protocolo. Por supuesto, el protocolo debe ser especificado primero con el comando iptables, mediante el uso de -p tcp <protocol-name> (donde <protocol-name> es el protocolo objetivo), para hacer las opciones disponibles para ese protocolo.

- **GUARDAR INFORMACIÓN DE IPTABLES**

Las reglas creadas con el comando iptables son almacenadas en memoria. Si el sistema es reiniciado después de configurar las reglas de iptables, se perderán. Para que las reglas de filtrado de red persistan luego de un reinicio

del sistema, estas necesitan ser guardadas. Para hacerlo, conéctese como root y escriba:

```
/sbin/service iptables save
```

Esto ejecuta el init script de iptables, el cual ejecuta el programa `/sbin/iptables-save` y escribe la configuración actual de iptables a `/etc/sysconfig/iptables`. Este archivo sólo debería estar accesible para root.

La próxima vez que se inicie el sistema, el script de inicio de iptables volverá a aplicar las reglas guardadas en `/etc/sysconfig/iptables` usando el comando `/sbin/iptables-restore`.

Aún cuando siempre es una buena idea probar una regla de iptables antes de confirmar los cambios al archivo `/etc/sysconfig/iptables`, es posible copiar reglas iptables en este archivo desde otra versión del sistema de este archivo. Esto proporciona una forma rápida de distribuir conjuntos de reglas iptables a muchas máquinas.

2.6.2 LOS WRAPPERS TCP Y EL COMANDO XINETD

El control del acceso a los servicios de red puede ser todo un reto.

Afortunadamente, bajo Red Hat Linux, hay un gran número de herramientas que lo pueden ayudar. Por ejemplo, un filtro de cortafuegos basado en iptables dejan afuera los paquetes de red que no son bienvenidos dentro de la pila de red del kernel. Para los servicios de red que lo utilizan, los *wrappers TCP* añaden una capa adicional de protección mediante la definición de cuáles hosts son permitidos y cuáles no conectarse a los

servicios de red "*wrapped*". Uno de los servicios de red wrapped es xinetd *super servidor*. Este servicio se le llama super servidor porque controla la conexión a un subconjunto de servicios de red y refina aún más el control de acceso.

o WRAPPERS TCP

El paquete wrappers TCP (tcp_wrappers) está instalado por defecto bajo Red Hat Linux y proporciona control de acceso basado en host a los servicios de red. El componente más importante dentro del paquete es la librería /usr/lib/libwrap.a. En términos generales, un servicio wrappers TCP es uno que ha sido compilado con la librería libwrap.a.

Cuando un intento de conexión es hecho a un servicio wrapped TCP, el servicio primero referencia los archivos de *acceso de host* (/etc/hosts.allow y /etc/hosts.deny) para determinar si el cliente tiene permitido conectarse. Luego utiliza el demonio syslog (syslogd) para escribir el nombre del host solicitante y el servicio solicitado a /var/log/secure o /var/log/messages.

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

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

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

o VENTAJAS DE LOS WRAPPERS TCP

Los wrappers TCP ofrecen las siguientes ventajas básicas comparadas con las otras técnicas de control de servicios de red:

- **Transparencia tanto para el cliente del host y el servicio de red wrapped.** El cliente que se está conectando así como también el servicio de red wrapped no están al tanto de que están en uso los wrappers TCP. Los usuarios legítimos son registrados y conectados al servicio solicitado mientras que las conexiones de clientes prohibidos fallan.
- **Administración centralizada** de protocolos múltiples. Los wrappers TCP operan separadamente de los servicios de red que ellos protegen, permitiendo a muchas aplicaciones de servidor compartir un conjunto común de archivos de configuración para una administración más sencilla.

o ARCHIVOS DE CONFIGURACIÓN DE WRAPPERS TCP

Para determinar si una máquina cliente tiene permitido conectarse a un servicio, los wrappers TCP referencian los siguientes dos archivos, los cuales se conocen comúnmente como archivos de acceso a host:

. /etc/hosts.allow

. /etc/hosts.deny

Cuando una solicitud de un cliente es recibida por un servicio wrapped TCP, sigue los pasos siguientes:

1. *El servicio referencia a /etc/hosts.allow.* . El servicio wrapped TCP analiza secuencialmente el archivo /etc/hosts.allow y aplica la primera regla especificada para ese servicio. Si encuentra una regla que coincide, permite la conexión. Si no, se va al paso 2.

2. *El servicio referencia /etc/hosts.deny.* . El servicio wrapped TCP analiza secuencialmente el archivo /etc/hosts.deny. Si encuentra una regla que coincide, rechaza la conexión. Si no, se concede acceso al servicio.

Lo siguiente son puntos muy importantes a considerar cuando se usen wrappers TCP para proteger servicios de red:

- Puesto que las reglas de acceso en hosts.allow son aplicadas primero, ellas toman precedencia sobre las reglas en hosts.deny. Por lo tanto, si se permite el acceso a un servicio en hosts.allow, una regla negando el acceso al mismo servicio en hosts.deny es ignorada.

- Puesto que las reglas en cada archivo son leídas de arriba hacia abajo y la primera regla que coincida para un servicio dado es la única aplicada, el orden de las reglas es extremadamente importante.
- Si no se encuentra ninguna regla para el servicio en ninguno de los archivos, o si no existe ninguno de los archivos, se concede el acceso al servicio.
- Los servicios wrapped TCP no hacen caché de las reglas desde los archivos acceso de host, por lo tanto cualquier cambio a hosts.allow o a hosts.deny tomarán efecto de inmediato sin tener que reiniciar el servicio de red.

o FORMATEAR REGLAS DE ACCESO

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

Las reglas se tienen que formatear de la siguiente manera:

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

<daemon list> Una lista separada por comas de los nombres de procesos (no de los nombres de servicios) o el comodín ALL.

<client list> Una lista separada por comas de nombres de host, direcciones IP, patrones especiales, o comodines especiales el cual identifica los hosts afectados por la regla.

<option> Una acción opcional o una lista separada con puntos y comas de acciones realizadas cuando la regla es activada. Los campos de opciones soportan *expansiones*, lance los comandos desde el shell, otorgue o prohíba el acceso y altere el comportamiento de registro.

A continuación una muestra básica de una regla de acceso:

```
vsftpd : .example.com
```

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

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

```
sshd : .example.com  \
: spawn  /bin/echo  `/bin/date` access denied>> /var/log/sshd.log  \
: deny
```

Note que en este ejemplo cada campo de opción está precedido por una barra oblicua invertida (\). Use la barra para prevenir la falla de la regla debido al largo.

Esta regla de ejemplo indica que si una conexión al demonio SSH (sshd) se intenta desde un host en el dominio example.com, ejecute el comando echo (lo cual registrará el intento a un archivo especial) y rechace la conexión.

Puesto que se usa la directiva opcional deny, esta línea rechazará el acceso aún si aparece en el archivo hosts.allow.

o **COMODINES**

Los comodines permiten a los wrappers TCP coincidir más fácilmente grupos de demonios o hosts. Son usados con mayor frecuencia en el campo de lista de cliente de las reglas de acceso.

Se pueden utilizar los siguientes comodines:

ALL. Hace corresponder todo. Se puede usar para la lista de demonios o en la lista de clientes.

LOCAL. Hace corresponder todos los nombres de máquinas que no contengan un punto (.), tal como localhost.

KNOWN Hace corresponder todas las máquinas cuyos nombres y direcciones son conocidos o donde el usuario es conocido.

UNKNOWN. Hace corresponder todas las máquinas cuyos nombres y direcciones sean desconocidas o en el caso en el que se desconozca el usuario.

PARANOID. Hace corresponder todas las máquinas cuyo nombre no se corresponda con la dirección.

o **OPERADORES**

Actualmente, las reglas de control de acceso aceptan un operador, EXCEPT. Se puede usar tanto en la lista de demonios como en la lista de cliente de una regla.

El operador EXCEPT permite excepciones específicas a coincidencias más amplias dentro de la misma regla.

En el ejemplo siguiente desde un archivo hosts.allow, todos los hosts de example.com pueden conectarse a todos los servicios excepto cracker.example.com:

```
ALL: .example.com EXCEPT cracker.example.com
```

En el otro ejemplo desde un archivo hosts.allow, clientes desde la red 192.168.0.x pueden usar todos los servicios excepto para FTP:

```
ALL EXCEPT vsftpd: 192.168.0.
```

o CAMPOS DE OPCIONES

Además de las reglas básicas para permitir o prohibir el acceso, la implementación de Red Hat Linux de wrappers TCP soporta extensiones al lenguaje de control de acceso a través de los campos de opciones. Mediante el uso de campos de opciones dentro de las reglas de acceso al host, los administradores pueden llevar a cabo una gran variedad de tareas tales como alterar el comportamiento del registro, consolidar el control de acceso y lanzar comandos del shell.

o REGISTRO

Los campos de opciones le permiten a los administradores cambiar fácilmente la facilidad de conexión y el nivel de prioridad para una regla usando la directiva severity.

En el ejemplo siguiente, las conexiones al demonio SSH desde cualquier host en el dominio example.com son registrados a la facilidad por defecto authpriv (debido a que no se especifica un valor de facilidad) con una prioridad de emerg:

```
sshd : .example.com : severity emerg
```

Es también posible especificar una facilidad utilizando la opción severity. El ejemplo siguiente registra cualquier intento de conexión SSH por cualquier hosts desde el dominio example.com a la facilidad local con una prioridad de alert:

```
sshd : .example.com : severity local0.alert
```

o CONTROL DE ACCESO

Los campos de opciones también le permiten a los administradores explícitamente otorgar o prohibir el acceso de máquinas en un sola regla, añadiendo la directiva allow o deny al final de la opción.

Por ejemplo, las dos reglas siguientes permiten conexiones SSH desde client-1.example.com, pero prohíben conexiones desde client-2.example.com:

```
sshd : client-1.example.com : allow
```

sshd : client-2.example.com : deny

Al permitir el control de acceso por regla, el campo de opciones permite a los administradores consolidar todas las reglas de acceso en un sólo archivo: bien sea hosts.allow o hosts.deny. Algunos consideran que esta es una forma más fácil de organizar reglas de acceso.

o COMANDOS DE LA SHELL

Los campos de opciones permiten a las reglas de acceso lanzar comandos de la shell a través de las directivas siguientes:

spawn . Lanza un comando de la shell como un proceso hijo. Esta directiva de opción puede realizar tareas como el uso de /usr/sbin/safe_finger para obtener más información sobre el cliente solicitante o la creación de archivos de registro especiales usando el comando echo.

En el ejemplo siguiente, los clientes intentando acceder servicios Telnet desde el dominio example.com son registrados discretamente a un archivo especial:

```
in.telnetd : .example.com \  
: spawn /bin/echo `/bin/date` from %h>>/var/log/telnet.log \  
: allow
```

twist.Reemplaza el servicio solicitado con el comando especificado. Esta directiva es a menudo usada para colocar trampas para intrusos (también llamados "potes de miel"). También se puede utilizar para enviar mensajes a

los clientes que se están conectando. El comando `twist` debe ocurrir al final de la línea de la regla.

En el ejemplo siguiente, los clientes intentando acceder servicios FTP desde el dominio `example.com` se les envía un mensaje a través del comando `echo`:

```
vsftpd : .example.com \
```

```
: twist /bin/echo "421 Bad hacker, go away!"
```

Para más información sobre las opciones de comando de la shell, consulte la página del manual de `hosts_options`.

o **XINETD**

El demonio `xinetd` es un *super servicio wrapped* TCP que controla el acceso a un subconjunto de servicios de red populares incluyendo FTP, IMAP y Telnet. También proporciona opciones de configuración específicas al servicio para el control de acceso, registro mejorado, re direccionamiento y control de utilización de recursos.

Cuando un host cliente intenta conectarse a un servicio de red controlado por `xinetd`, el súper servicio recibe la petición y verifica por cualquier regla de control de acceso wrappers TCP. Si se permite el acceso, `xinetd` verifica que la conexión sea permitida bajo sus propias reglas para ese servicio y que el servicio no esté consumiendo más de la cantidad de recursos o si está rompiendo alguna regla. Luego comienza una instancia del servicio solicitado y pasa el control de la conexión al mismo. Una vez establecida la conexión,

xinetd no interfiere más con la comunicación entre el host cliente y el servidor.

o EL ARCHIVO /ETC/XINETD.CONF

El archivo `/etc/xinetd.conf` contiene parámetros de configuración generales los cuales afectan cada servicio bajo el control de xinetd. Se lee una vez cuando el servicio xinetd es iniciado, por esto para que los cambios de la configuración tomen efecto, el administrador debe reiniciar el servicio xinetd.

Abajo se muestra un ejemplo del archivo `/etc/xinetd.conf`:

```
defaults
{
    instances            = 60
    log_type              = SYSLOG authpriv
    log_on_success        = HOST PID
    log_on_failure        = HOST
    cps                  = 25 30
}
includedir /etc/xinetd.d
```

Estas líneas controlan varios aspectos de xinetd:

instances -Configura el máximo número de peticiones que xinetd puede manejar simultáneamente.

log_type - Configura xinetd para usar la facilidad de registro authpriv, el cual escribe las entradas de registro al archivo `/var/log/secure`. Al agregar una directiva tal como `FILE /var/log/xinetdlog` aquí, crear á un archivo de registro personalizado llamado xinetdlog en el directorio `/var/log/`.

log_on_success. -Configura xinetd a registrar si la conexión es exitosa. Por defecto, la dirección IP del host remoto y el ID del proceso del servidor procesando la petición son grabados.

log_on_failure-Configura xinetd para registrar si hay una falla de conexión o si la conexión no es permitida.

cps.-Configura xinetd para no permitir más de 25 conexiones por segundo a cualquier servicio dado. Si se alcanza este límite, el servicio es retirado por 30 segundos.

includedir /etc/xinetd.d/ -Incluye las opciones declaradas en los archivos de configuración específicos del servicio localizados en el directorio /etc/xinetd.d/.

o MODIFICAR ARCHIVOS DE CONFIGURACIÓN XINETD

Existe una gran cantidad de directivas disponibles para los servicios xinetd protegidos. Esta sección resalta algunos de las opciones usadas más comúnmente.

o OPCIONES DE REGISTRO

Las siguientes opciones de registro están disponibles para /etc/xinetd.conf y los archivos de configuración específicos al servicio en el directorio /etc/xinetd.d/.

Abajo se muestra una lista de algunas de las opciones de registro usadas más comúnmente:

ATTEMPT. -Indica que se intentó realizar una conexión pero que ésta falló (log_on_failure).

DURATION. -Indica el tiempo que un sistema remoto usa un servicio (log_on_success).

EXIT.-Indica el estado de salida o la señal de término del servicio (log_on_success).

HOST. -Indica la dirección IP de la máquina remota (log_on_failure y log_on_success).

PID. -Indica el ID del proceso del servidor que recibe la petición (log_on_success).

RECORD. -Graba información sobre el sistema remoto en el caso de que no se pueda iniciar el servicio. Esta opción la usan solo determinados tipos de servicios como login y finger, puede usar esta opción (log_on_failure).

USERID.Registra el usuario remoto que está usando el método definido en RFC 1413 para todos los servicios de multi procesos (log_on_failure y log_on_success).

o OPCIONES DE CONTROL DE ACCESO

Los usuarios de servicios xinetd pueden seleccionar usar reglas de acceso a hosts de wrappers TCP, proporciona control de acceso a través de los archivos de configuración xinetd, o una mezcla de ambos. La información concerniente al uso de los archivos de control de acceso a hosts wrappers TCP

El control de acceso xinetd es diferente del método usado por los wrappers TCP. Mientras que los wrappers TCP colocan toda la configuración del acceso dentro de dos archivos, /etc/hosts.allow y /etc/hosts.deny, el archivo de cada servicio en /etc/xinetd.d puede contener sus propias reglas de control de acceso.

Las opciones de acceso a host siguientes son soportadas por xinetd:

only_from. -Sólo permite que las máquinas específicas usen el servicio.

no_access. -Impide que estas máquinas usen el servicio.

access _ times . -Especifica el intervalo de tiempo en el que un determinado servicio puede ser usado. El rango de tiempo debe especificarse en formato de 24 horas, HH:MM-HH:MM.

Las opciones only_from y no_access pueden usar una lista de direcciones IP o nombres de hosts, o pueden especificar una red completa. Como los wrappers TCP, la combinación del comando xinetd para el control del acceso con una configuración de conexión apropiada puede mejorar la seguridad

mediante el bloqueo de peticiones de hosts vetados mientras que graba cada intento de conexión.

Por ejemplo, el siguiente archivo `/etc/xinetd.d/telnet` puede ser usado para bloquear el acceso a Telnet desde un grupo de red particular y restringir el rango de tiempo general que inclusive los usuarios permitidos pueden conectarse: `service telnet`

```
{
    disable                = no
    flags                   = REUSE
    socket_type             = stream
    wait                   = no
    user                   = root
    server                  = /usr/sbin/in.telnetd
    log_on_failure +       = USERID
    no_access               = 10.0.1.0/24
    log_on_success +       = PID HOST EXIT
    access_times            = 09:45-16:15
}
```

En este ejemplo, cuando un sistema cliente desde la red 10.0.1.0/24, tal como 10.0.1.2, intenta acceder el servicio Telnet, recibirá un mensaje indicando lo siguiente:

Connection closed by foreign host.

Además, su intento de conexión es registrado en `/var/log/secure` como sigue:

May 15 17:38:49 boo xinetd[16252]: START: telnet pid=16256 from=10.0.1.2

May 15 17:38:49 boo xinetd[16256]: FAIL: telnet address from=10.0.1.2

May 15 17:38:49 boo xinetd[16252]: EXIT: telnet status=0 pid=16256

Cuando esté usando wrappers TCP en conjunto con controles de acceso xinetd, es importante entender la relación entre los dos mecanismos de control de acceso.

A continuación se muestra el orden de las operaciones seguido por xinetd

Cuando un cliente solicita una conexión:

1. El demonio xinetd accesa las reglas de acceso a hosts wrappers TCP a través de una llamada a la librería libwrap.a. Si alguna regla de rechazo coincide con el host cliente, la conexión se rechaza. Si una regla de aceptación coincide con el host cliente, la conexión es pasada al xinetd.
2. El demonio xinetd verifica sus propias reglas de acceso para el servicio xinetd y el servicio solicitado.

CAPITULO III PROYECTO TEMATICO

3.1 PLANTEAMIENTO DEL PROYECTO TEMÁTICO

Según un estudio realizado previamente a la carrera de Técnico en Ingeniería de Redes Computacionales, se ve reflejado un buen porcentaje de desconocimiento de estrategias de seguridad en el sistema Linux, orientado a una aplicación de servicios de Internet y al mismo tiempo se demostró que un buen porcentaje de alumnos de la carrera sienten un interés por conocer más sobre estrategias de seguridad a nivel de servidor de servicio de Internet.

Por tal razón se dio la necesidad de crear un manual técnico de aplicaciones de estrategias de seguridad a nivel de servidor de servicio de Internet, que contenga tanto las vulnerabilidades como sus soluciones y al mismo tiempo aporte algunas ayudas para reforzar la seguridad.

La creación del manual técnico consta de las siguientes fases:

Fase 1

El manual técnico presenta una selección de algunas vulnerabilidades que afectan al sistema operativo y por consiguiente, son aprovechadas por los hackers, cracker u otro sistema malicioso, para así poder llevar acabo sus malévolos planes.

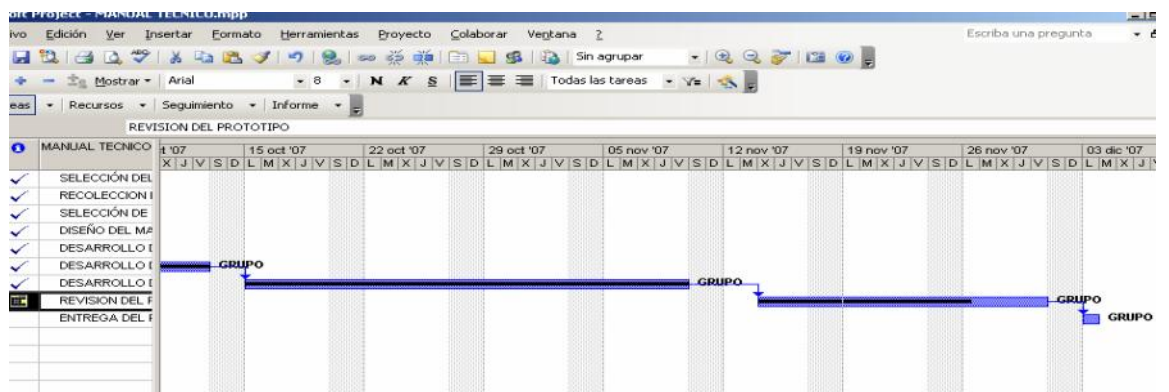
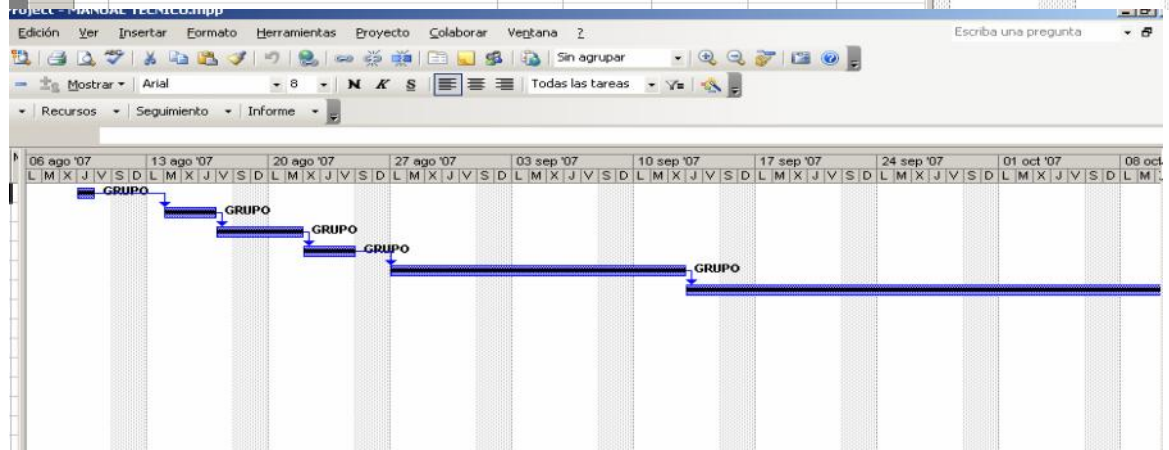
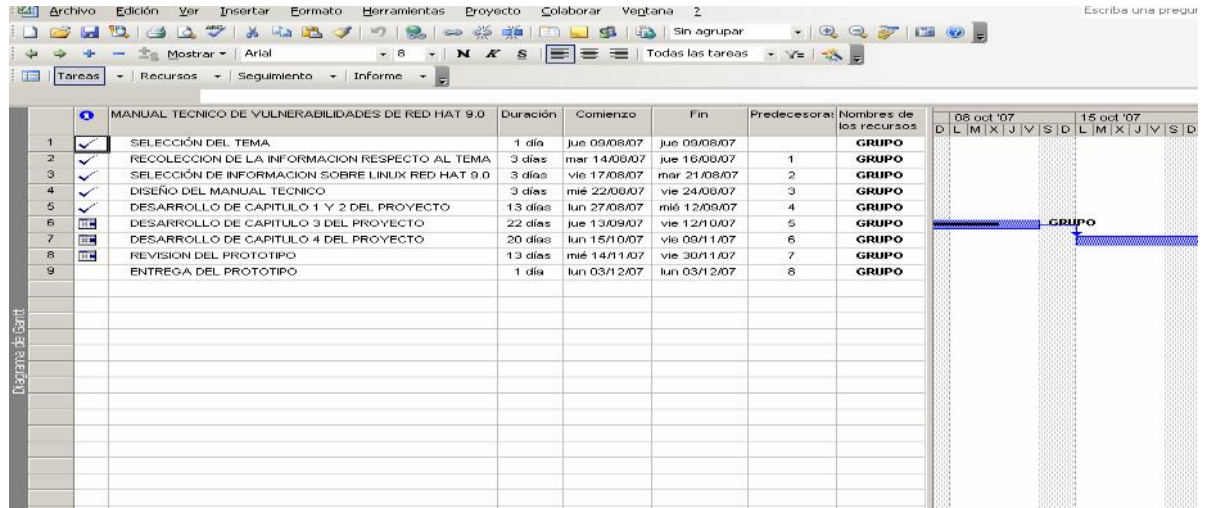
Fase 2

Al mismo tiempo el manual aportara soluciones a algunas de las vulnerabilidades presentadas y otras no presentadas, las cuales algunas de ellas no son de configuración sino de prevención a nivel físico.

Fase 3

También el manual aporta ideas y configuraciones para poder aumentar así la seguridad de un servidor que brinda el servicio de Internet

3.2 CRONOGRAMA DEL PROYECTO



3.3 DOCUMENTACIÓN TECNICA

- **VERSIONES DE LINUX**

- **LINUX RED HAT 9.0**

Red Hat es una distribución Linux creada por Red Hat, que fue una de las más populares en los entornos de usuarios domésticos.

Es una de las distribuciones Linux de "mediana edad". La versión 1.0 fue presentada el 3 de Noviembre de 1994. No es tan antigua como la distribución Slackware, pero ciertamente es más antigua que muchas otras. Fue la primera distribución que usó RPM como su formato de paquete, y en un cierto plazo ha servido como el punto de partida para varias otras distribuciones, tales como la orientada hacia PCs de escritorio Mandrake Linux (originalmente Red Hat Linux con KDE), Yellow Dog Linux, la cual se inició desde Red Hat Linux con soporte para PowerPC, y ASPLinux (Red Hat Linux con mejor soporte para caracteres no-Latinos).

Desde el 2003, Red Hat ha desplazado su enfoque hacia el mercado de los negocios con la distribución Red Hat Enterprise Linux y la versión no comercial Fedora Core. Red Hat Linux 9, la versión final, llegó oficialmente al final de su vida útil el pasado 30 de abril de 2004, aunque el proyecto Fedora Legacy continuó publicando actualizaciones, hasta ser abandonado dicho proyecto a finales de 2006.

REQUERIMIENTOS MINIMOS PARA INSTALACIÓN

Instalación de tipo servidor

Una instalación de tipo servidor es más apropiada si quiere que su sistema funcione como un servidor de tipo Linux y no desea personalizar mucho su sistema.

A continuación tiene los requisitos mínimos de espacio en el disco para una instalación de tipo servidor en la que tan sólo se instalará un idioma (por ejemplo Español).

Servidor (mínimo, sin interfaz gráfica): 850 MB

Servidor (seleccionando todo, sin interfaz gráfica): 1.5 GB

Servidor (seleccionando todo, incluyendo una interfaz gráfica): 5.0 GB

(Ver URL en anexo pag. #142)

➤ **UBUNTU**

- **Ubuntu** (AFI: /u'buntu/) es una distribución Linux que ofrece un sistema operativo predominantemente enfocado a ordenadores de escritorio aunque también proporciona soporte para servidores.

Basada en Debian GNU/Linux, Ubuntu concentra su objetivo en la facilidad de uso, la libertad de uso, los lanzamientos regulares (cada 6 meses) y la facilidad en la instalación. Ubuntu es patrocinado por Canonical Ltd., una

empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.

El nombre de la distribución proviene del concepto zulú y xhosa de *ubuntu*, que significa "humanidad hacia otros" o "yo soy porque nosotros somos". Ubuntu es un movimiento sudafricano encabezado por el obispo Desmond Tutu, quien ganó el Premio Nobel de la Paz en 1984 por sus luchas en contra del *Apartheid* en Sudáfrica. El sudafricano Mark Shuttleworth, mecenas del proyecto, se encontraba muy familiarizado con la corriente. Tras ver similitudes entre los ideales de los proyectos GNU, Debian y en general con el movimiento del software libre, decidió aprovechar la ocasión para difundir los ideales de *Ubuntu*. El eslogan de Ubuntu – “Linux para seres humanos” (en inglés "Linux for Human Beings") – resume una de sus metas principales: hacer de Linux un sistema operativo más accesible y fácil de usar.

REQUERIMIENTOS MINIMOS PARA INSTALACIÓN

	 DESKTOP	 SERVER
Memoria RAM mínima	256 MB	64 MB
Espacio en disco mínimo	3 GB	500 MB
Entorno de trabajo instalado	Gráfico	Modo texto
Kernel	Estándar	Varios, optimizados
Servidor web	Disponible	Disponible, simplificado
Soporte de servicios de red	Estándar	Mejorado
Soporte de clusters y SANs	Estándar	Mejorado
Años de soporte técnico LTS	3 años	5 años

(Ver URL anexo en pag. #142)

➤ DEBIAN

Debian o **Proyecto Debian**¹ (en inglés ***Debian Project***²) es una comunidad conformada por desarrolladores y usuarios, que pretende crear y mantener un sistema operativo GNU basado en software libre precompilado y empaquetado, en un formato sencillo en múltiples arquitecturas de computador y en varios núcleos.

Debian nace como una apuesta por separar en sus versiones el software libre del software no libre. El modelo de desarrollo del proyecto es ajeno a motivos empresariales o comerciales, siendo llevado adelante por los propios

usuarios, aunque cuenta con el apoyo de varias empresas en forma de infraestructuras. **Debian** no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia.

La comunidad de desarrolladores de Debian cuenta con la representación de **Software in the Public Interest**³, una organización sin ánimo de lucro que da cobertura legal a varios proyectos de software libre.

La primera adaptación del *sistema Debian*, siendo también la más desarrollada, es **Debian GNU/Linux**, basada en el núcleo Linux, y como siempre utilizando herramientas de GNU. Existen también otras adaptaciones con diversos núcleos: Hurd (Debian GNU/Hurd); NetBSD (Debian GNU/NetBSD) y FreeBSD (Debian GNU/kFreeBSD).

Requisitos Mínimos de Hardware

Una vez que haya reunido información sobre el hardware de su ordenador debe verificar que su hardware le permita realizar el tipo de instalación que desea efectuar.

Dependiendo de sus necesidades, podría arreglarse con menos del hardware recomendado listado en la siguiente tabla. Sin embargo, la mayoría de usuarios se arriesgan a terminar frustrados si ignoran estas sugerencias.

Se recomienda como mínimo un Pentium 4, a 1 GHz para un sistema de escritorio.

Tabla 3.2. Requisitos mínimos de sistema recomendados

Tipo de instalación	RAM (mínimo)	RAM (recomendado)	Disco duro
Sin escritorio	64 Megabytes	256 Megabytes	1 Gigabyte
Con escritorio	64 Megabytes	512 Megabytes	5 Gigabytes

Los requisitos de memoria mínimos necesarios son en realidad inferiores a los indicados en esta tabla. En función de la arquitectura, es posible instalar Debian en sistemas con tan sólo 20 MB (en el caso de s390) a 48 MB (para i386 y amd64). Lo mismo se puede decir del espacio necesario en disco, especialmente si escoge las aplicaciones que va a instalar manualmente, consulte Sección D.2, “Espacio en disco requerido para las tareas” para obtener más información de los requisitos de disco.

Es posible ejecutar un entorno de escritorio gráfico en sistemas antiguos o de gama baja. En este caso es recomendable instalar un gestor de ventanas que consuma menos recursos que los utilizados en los entornos de escritorio de GNOME o KDE. Algunas alternativas para estos casos son xfce4, icewm y wmaker, aunque hay más entre los que puede elegir.

Es prácticamente imposible dar requisitos generales de memoria y espacio en disco para instalaciones de servidores ya que éstos dependerán en gran medida de aquello para lo que se utilice el servidor.

Recuerde que estos tamaños no incluyen todos los otros materiales que se encuentran habitualmente, como puedan ser los ficheros de usuarios, el correo y otros datos. Siempre es mejor ser generoso cuando uno está pensando qué espacio destinar a sus propios ficheros y datos.

Se ha tenido en cuenta el espacio de disco necesario para la operación normal del sistema Debian GNU/Linux en sí en estos requisitos de sistema recomendados. En particular, la partición /var de Debian contiene mucha información de estado específica a Debian, además de su contenido habitual, como puedan ser los ficheros de registro. Los ficheros de **dpkg** (que incluyen información sobre los paquetes instalados) pueden fácilmente consumir unos 40 MB. Además, hay que tener en cuenta que **apt-get** ubica los paquetes descargados aquí antes de instalarlos. Por regla general deberá asignar por lo menos 200 MB para /var, y mucho más si va a instalar un entorno gráfico de escritorio.

(Ver URL en pag. # 143)

- **EDITORES DE TEXTO**

- ✓ **ADOBE ACROBAT READER 8**

Nueva interfaz, nuevas herramientas y más opciones

Saque el máximo partido al espacio de su pantalla Reader 8 ha rediseñado completamente la interfaz y presenta nuevas herramientas y una amplia gama de opciones nuevas para visualizar la información de forma más eficaz.

Acceda a una reunión en línea y tiempo real en segundos

Reader 8 está integrado ahora en el software Adobe Connect, que permite a los usuarios comunicarse de forma instantánea y acelerar las aprobaciones prácticamente con cualquiera, en cualquier lugar, a cualquier hora.

Mejor protección para sus flujos de trabajo y transacciones

Añada más seguridad a sus documentos, formularios y dibujos con las firmas electrónicas. Aproveche Adobe LiveCycle® Policy Server y el nuevo centro de documentación de servicios en línea con Reader 8 para reforzar los derechos de acceso a los documentos .

(Ver URL en pag. # 143)

✓ **DREAMWEAVER CS3**

Diseñe, desarrolle y realice el mantenimiento integral de sitios y aplicaciones web de forma rápida y sencilla con Adobe Dreamweaver CS3. Creado para diseñadores y desarrolladores, Dreamweaver CS3 ofrece la posibilidad de trabajar en una interfaz de diseño visual intuitiva o en un entorno de codificación agilizado. La integración inteligente con el software Adobe Photoshop® CS3, Adobe Illustrator® CS3, Adobe Fireworks® CS3, Adobe Flash CS3 Professional y Adobe Contribute® CS3 le garantiza un flujo de trabajo eficaz con sus herramientas favoritas.

(VER ANEXO PAG. #144)

✓ **POWER POINT 2003**

Trabaje mejor con otros usuarios: puede guardar las presentaciones de PowerPoint 2003 en áreas de trabajo compartidas, donde otros miembros del equipo pueden obtener la versión más reciente y proteger o desproteger las presentaciones. Con las áreas de trabajo compartidas, puede guardar listas de tareas, archivos relacionados, vínculos y listas de miembros que trabajan en un proyecto. Las áreas de trabajo compartidas requieren el uso de Microsoft Windows Server™ 2003 con Microsoft Windows SharePoint™ Services.

Revise con su equipo: puede crear un Área de documentos mediante la opción Datos adjuntos compartidos al adjuntar la presentación a un mensaje de correo electrónico de Microsoft Office Outlook 2003. Después, el equipo puede utilizar un panel de tareas en la presentación para colaborar en ella y revisarla. Las áreas de trabajo compartidas requieren el uso de Windows Server 2003 con Windows SharePoint Services.

Controle la distribución de las presentaciones: para proteger los activos de la compañía, puede impedir que los destinatarios reenvíen, copien o impriman presentaciones importantes mediante la funcionalidad de Administración de derechos de información (IRM, Information Rights Management). Puede especificar una fecha de caducidad para una presentación, tras la que no se podrá ver o cambiar el documento. La funcionalidad IRM requiere el uso de Windows Server 2003 con Microsoft Windows Rights Management Services (RMS).

Nota con Microsoft Office Professional Edition 2003, puede utilizar PowerPoint para crear presentaciones protegidas por IRM y conceder a otros usuarios permiso para tener acceso a las presentaciones y modificarlas. También puede aplicar plantillas de directivas a las presentaciones protegidas por IRM que cree. Con Microsoft Office Standard Edition 2003, puede leer presentaciones protegidas por IRM; con los permisos, también puede modificarlas.

(Ver URL en pag de anexo #144)

3.4 EVALUACIÓN TÉCNICA Y ECONÓMICA

EVALUACIÓN TÉCNICA DE DISTRIBUCIONES DE LINUX

Tabla comparativa de distribuciones

Red Hat 9.0

Ubuntu 6.06 LTS

Debían 4.0

[illegible]

Tabla comparativa de Editores de Texto								
			Adobe Acrobat 8.0		Dreanwaeaver		Microsoft Office PowerPoint 2003	
Características	Valor requerido	Peso ó porcentaje	Ponderación	Calificación	Ponderación	Calificación	Ponderación	Calificación
Funcionalidad	Facilidad de uso.	15%	2	15%	1	7,50%	2	15%
Seguridad	Seguridad en la integridad del documento.	35%	2	35%	1	17.5%	0	0%
Adquisición del producto	Facilidad de adquisición.	20%	2	20%	1	10%	1	10%
Compatibilidad de versiones	Compatibilidad con versiones anteriores.	30%	2	30%	1	15%	1	15%
Total				100%		50%		40%
Ponderación			* El valor para elegir y utilizar la distribución de los editores de texto deberá obtener un porcentaje de 100%					
0 = No cumple las especificaciones								
1 = Cumple en un 50%, pero es factible								
2 = Sobrepasa las expectativas								

EVALUACIÓN TÉCNICA DE EDITORES DE TEXTO

EVALUACIÓN ECONÓMICA

EVALUACIÓN ECONÓMICA

Para el desarrollo del manual técnico para el laboratorio de redes se requiere de una evaluación económica, concisa para llegar a determinar que editor puede ser más accesible.

EVALUACIÓN ECONÓMICA		
Editores de texto	Valor	Editor SELECCIONADO
Adobe Acrobat Reader 8.0	DEMO	X
Dreanwaeaver CS3 (full)	\$ 400,00	
Microsoft Office Power Point 2003	\$ 195,00	

La selección del editor de texto más apropiada para el desarrollo del manual es **ADOBE ACROBAT READER 8,0**

3.5 TECNOLOGÍAS Y RECURSOS SELECCIONADOS

TECNOLOGÍA Y RECURSOS SELECCIONADOS	
Linux Red Hat 9,0	El sistema operativo Red Hat 9.0, cumple con los requisitos anteriormente presentados en la comparación con el sistema operativo Windows 2003 Server el cual puede ser implementado en el laboratorio de redes de la UTEC.
RECURSOS ADICIONALES	
ADOBE ACROBAT VERSIÓN 8	Para mayor seguridad y presentación del manual técnico utilizaremos este software como herramienta para la edición del manual.
ARCHIVO EJECUTABLE	Servirá para mayor facilidad de presentación de este manual técnico.

3.6 PRESUPUESTO PROYECTADO

Para la creación del manual técnico se requiere de recursos, que son indispensables para el desarrollo del manual, se necesita de una inversión económica como también de recurso humano; estos se ven reflejados en el siguiente cuadro:

RECURSOS A UTILIZAR		
RECURSOS	DESCRIPCIÓN	COSTOS
200	Fotocopias (hojas) a un costo (0,2 Cts.)	\$4,00
24 horas	Internet, a un costo \$ 1,00 (aproximado)	\$45,00
400	Impresiones, a un costo (\$ 0,07) como mínimo	\$28,00
1	Computadora Pentium III con 256 de RAM, de escritorio	\$250,00
1	Acrobat versión 8,0 (software)	Versión DEMO
	TOTAL	\$ 327,00



3.7 OFERTA TÉCNICA

TUX SA DE CV

San Salvador 16 de Noviembre de 2007

Señores:

ALFA EKO

Presente

Atención: In. Juan Polanco

Estimado; Ing. Polanco

Tenemos el agrado de presentar el nuevo Manual Técnico de Estrategias de Seguridad a Nivel de Servidor de Servicios de Internet basado en Linux Red Hat 9.0, de la empresa **TUX SA DE CV**, el cual tiene la intención de brindarle a su empresa una selección de estrategias de seguridad, para una mayor protección de su servidor de servicios de Internet.

TUX SA DE CV, tiene como base fundamental ofrecerle estrategias de seguridad; para una mejor protección de su servidor contra algunas amenazas tales como: Hackers, Cracker, Troyanos, gusanos, etc, que intentan acceder al sistema para hacer uso ó dañar el sistema operativo de red.

El manual técnico de estrategias de seguridad, prevendrá de costos adicionales en la compra de equipo dañado, o información robada a su empresa. Ya que con el uso de las configuraciones expuestas en el manual técnico, le dará la seguridad necesaria para la protección de sus datos y servicios de red.

Esperando tener la oportunidad de poder participar en la solución de sus necesidades y sin más por el momento, nos suscribimos cordialmente.

Franklin Alberto Zelada Escobar

Técnico en Sistemas de Redes Informáticos

TEL. 2517-6902

3.8 OFERTA ECONÓMICA

Para poder implementar el manual técnico de estrategias de seguridad a nivel de un servidor de servicio de Internet basado en Linux, el cual se implementará en el laboratorio de Redes de la Universidad Tecnológica de El Salvador se requiere de la adquisición del sistema operativo Linux Red Hat 9.0 y la compra del manual Técnico para poder así desarrollar las estrategias de seguridad expuestas en el manual.

RECURSOS NECESARIO PARA IMPLEMENTAR EL MANUAL	
	COSTO
Sistema Operativo Linux Red Hat 9.0	GRATUITO
Manual Técnico de Estrategias de Seguridad	\$ 350.00
Instalación y Configuración del sistema operativo en un servidor (mano de obra)	\$ 250.00
Soporte Técnico (garantía de 2 años)*	\$ 500.00

* La garantía del manual consiste en las configuraciones hechas en el sistema operativo instalado en el servidor, no incluye fallas al equipo (hardware).

CAPITULO IV PROPUESTA

4.1 ARTICULO DE MANUAL TÉCNICO DE ESTRATEGIAS DE SEGURIDAD PARA SERVIDORES DE SERVICIO DE INTERNET CON LINUX RED HAT 9.0

Dentro del medio estudiantil de la Universidad Tecnológica de El Salvador se encontró un desconocimiento del sistema operativo Linux, lo cual crea a los estudiantes de la carrera Técnico en Ingeniería de Redes Computacionales una deficiencia en el ámbito profesional, y por ende un menor desempeño y menores oportunidades de trabajo por falta del desconocimiento de este sistema operativo.

Por esta razón surgió la idea de la creación de un manual técnico, para reforzar los conocimientos a los estudiantes de la carrera Técnico en Ingeniería de Redes Computacionales de tener información relacionada con este sistema y específicamente en la seguridad del mismo, de tal forma se elaboro un manual técnico que aportara información sobre estrategias de seguridad para el área de un servidor de servicio de Internet.

Con esto, se lograra que los estudiantes tengan una ayuda en el desarrollo de su carrera técnica ya que contarán con un material de apoyo en sus prácticas en el laboratorio de redes.

Primeramente para el desarrollo del manual técnico, se llevo a cabo una búsqueda de información en distintos medios como: Revistas, Biblioteca

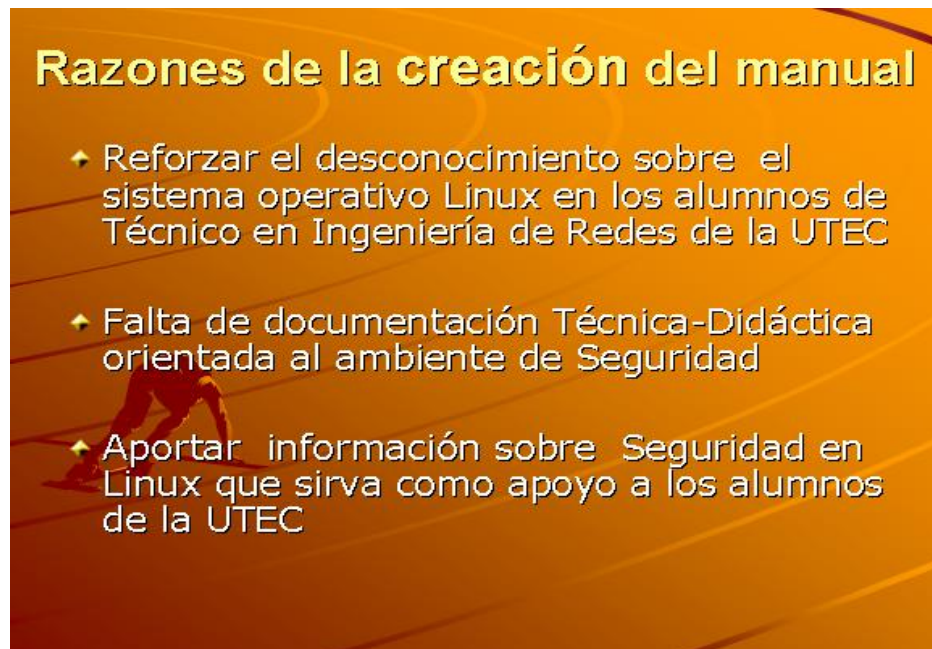
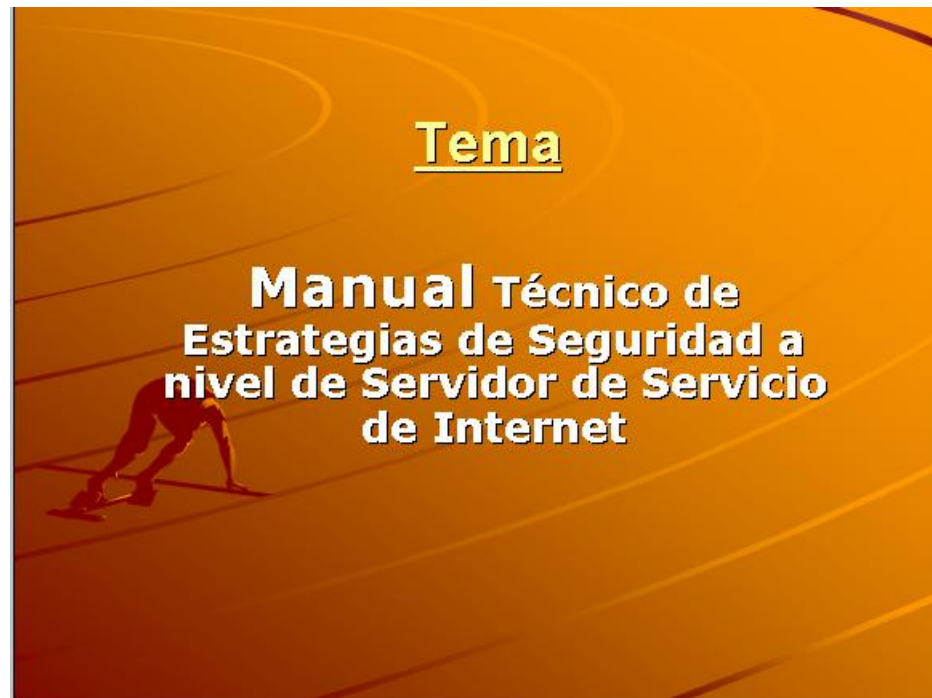
Central de la UTEC e Internet. Tomando como referencia la seguridad para su respectiva clasificación del material encontrado.

Una vez clasificada la información encontrada el siguiente paso consistió en la instalación del sistema operativo para realizar las configuraciones pertinentes de los parámetros que incrementan el nivel de seguridad en un servidor de servicio para acceso de Internet.

Obteniendo como resultado un mayor rendimiento, estabilidad en el entorno de la seguridad en comparación con un sistema configurado tradicionalmente, comprobando así que el manual técnico tiene los elementos necesarios para una fácil configuración y con un mayor nivel de seguridad; cumpliendo de esta manera el propósito para el cual fue creado.

El manual técnico aporta nuevos conocimientos los estudiantes de la carrera Técnico de Redes Computacionales, de esta forma brinda una motivación al estudiante para conocer y poner en práctica las aplicaciones expuestas .

4.2 PRESENTACIÓN DEL PROTOTIPO



Recopilación de información

La información utilizada para la creación del Manual Técnico fue obtenida de las siguientes fuentes:

- Biblioteca Central (Libros, Tesis, etc.)
- Internet
- Revistas

Puntos clave para la creación del Manual Técnico

- ◆ Recopilación de información
- ◆ Selección de la información
- ◆ Desarrollo del Manual Técnico
- ◆ Configuraciones
- ◆ Verificación de resultados obtenidos

Selección de la Información

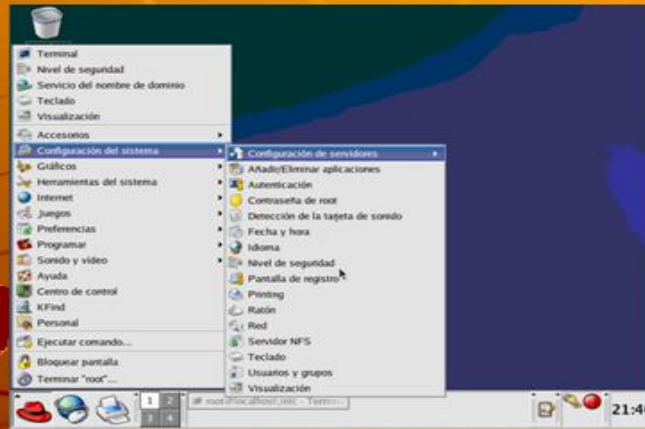
La selección de la información se efectuó bajo los siguientes parámetros

- o Sistema operativo Red Hat 9.0
- o Seguridad en ambiente de Servidor para Servicios de Internet

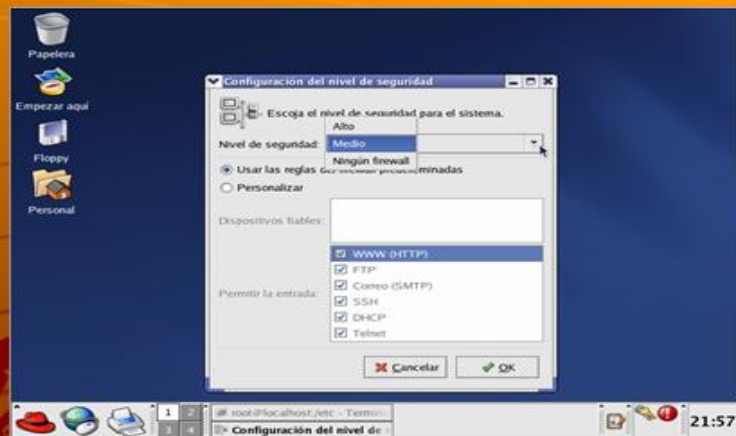
Desarrollo del Manual

- ◆ Instalación del sistema operativo
- ◆ Realizar configuraciones pertinentes que incrementan la seguridad en un Servidor de Servicios de Internet
- ◆ Pruebas de configuración

CONFIGURACIONES



- ✦ Para habilitar el firewall tendremos que ir a la barra de inicio del sistema, luego seleccionar la opción de configuraciones del sistema, después ir a nivel de seguridad.



- ✦ Como siguiente paso asignaremos el grado de seguridad para nuestro firewall

Beneficios obtenidos

- ◆ Documentación disponible para la carrera técnica de Redes
- ◆ Facilidad de uso del manual
- ◆ Aportación de nuevos conocimientos a los estudiantes de la carrera técnica de Redes
- ◆ Elementos necesarios para una fácil configuración que brindara una mayor seguridad

Resultados obtenidos

- Estabilidad en el entorno de seguridad
- Mayor rendimiento y funcionalidad

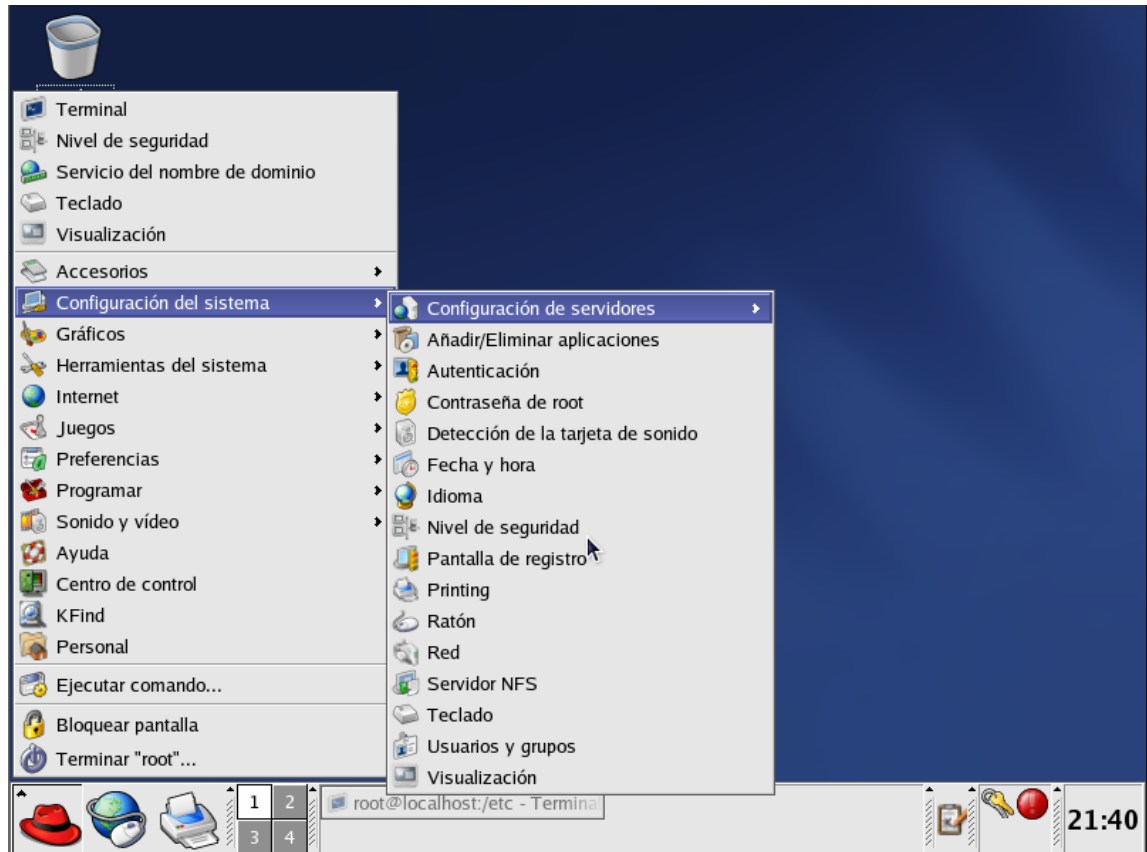
4.3 CONFIGURACIONES DE SEGURIDAD PARA EL SERVIDOR.

MENÚ DE CONFIGURACIONES BÁSICAS PARA SERVIDOR CON LINUX RED HAT 9,0		
AMENAZAS	VULNERABILIDADES	CONFIGURACIONES
TROYANO RST,b.	DESCARGA SIN PRECAUCIÓN	a) Ejecútelo antes en un entorno chroot cerrado. b) Nunca ejecute nada como root. c) Recomendación, todo programa poco fiable es mejor probarlo antes en una maquina virtual (WMWare), para comprobar si hace sospechoso antes de ejecutarlo en nuestro sistema.
Archivo tar	Extraer un <i>paquete tar</i> hay que ejecutar el programa tar con la opción "x" como, por ejemplo tar xvf tarball.tar o tar xzvf tarball en el caso de estar comprimido.	a) No extraer nunca los paquetes tar como root b) Asegúrese de comprobar posteriormente los ficheros suid y said
Paquetes rpm corruptos.	El programa rpm presentaba un error que podía provocar un desbordamiento de la pila si se la hacia una consulta a un paquete RPM malformado.	Actualice su programa RMP.
Ataque por desbordamiento de buffer.	Falta de una función segura de concatenación de cadenas.	Root # chmod u – s / usr / bin / sudo
Ataques al núcleo del sistema.	Error del núcleo de nuestro sistema.	La solución más segura para solucionar un error del núcleo de nuestro sistema pasa por actualizarlo.

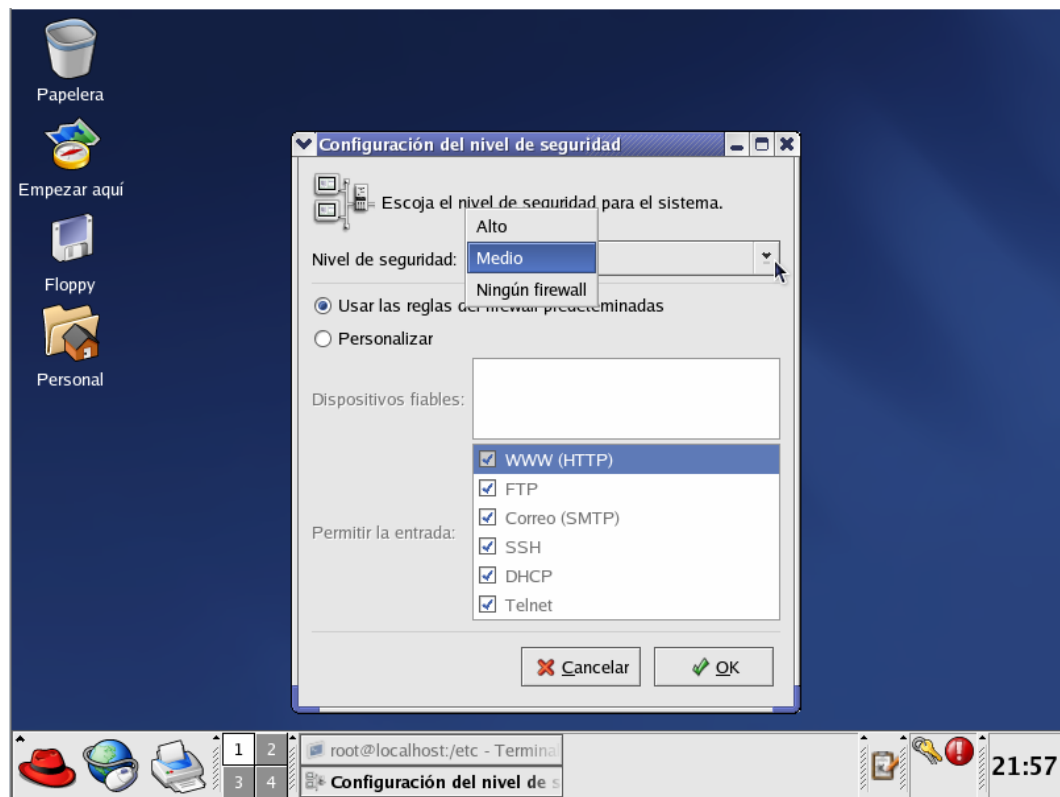
Acceso libre al servidor y recurso de red.	Libre entrada al sistema.	(FIREWALL)En la barra del menú KDE en la opción CONFIGURACIÓN DEL SISTEMA, luego ir a NIVEL DE SEGURIDAD, y habilitar el nivel de seguridad que se necesita.
Acceso ilegal al sistema.	XINETD.	(XINETD desde modo consola) En el directorio /etc/xinetd.conf se encuentra la opción de no acceso a una determinada maquina.
Ataque de agotamiento de recurso.	XINETD.	(XINETD)En el directorio /etc/xinetd.conf se encuentra la opción "DEFAULT" en el cual se encuentra de opción "INSTANCES" en donde se puede asignar el número de conexiones simultáneas a un servicio, "PER_SOURCE" en donde se puede asignar el número de conexiones simultáneas a un único servicio desde una única dirección IP.
Forzamiento del servicio de TELNET.	XINETD.	(XINETD desde modo consola) En el directorio /etc/hosts.allow y en el directorio /etc/hosts.deny se encuentran la opción del servicio de TELNET para los usuarios en donde se puede ver y configurar el acceso a la red.
Robo de información o contraseña, cuando el usuario se ausenta de su maquina por periodo corto de tiempo.	Lock screen.	Acceder al botón de inicio de red hat  , en la barra de herramientas seleccionar la opción de BLOQUEAR PANTALLA, luego solo se habilitara cuando sea necesario, después solo necesitaremos ingresar nuestra contraseña y volveremos a la sesión.

En esta parte del manual demostraremos algunas configuraciones de seguridad que se pueden realizar en el servidor.

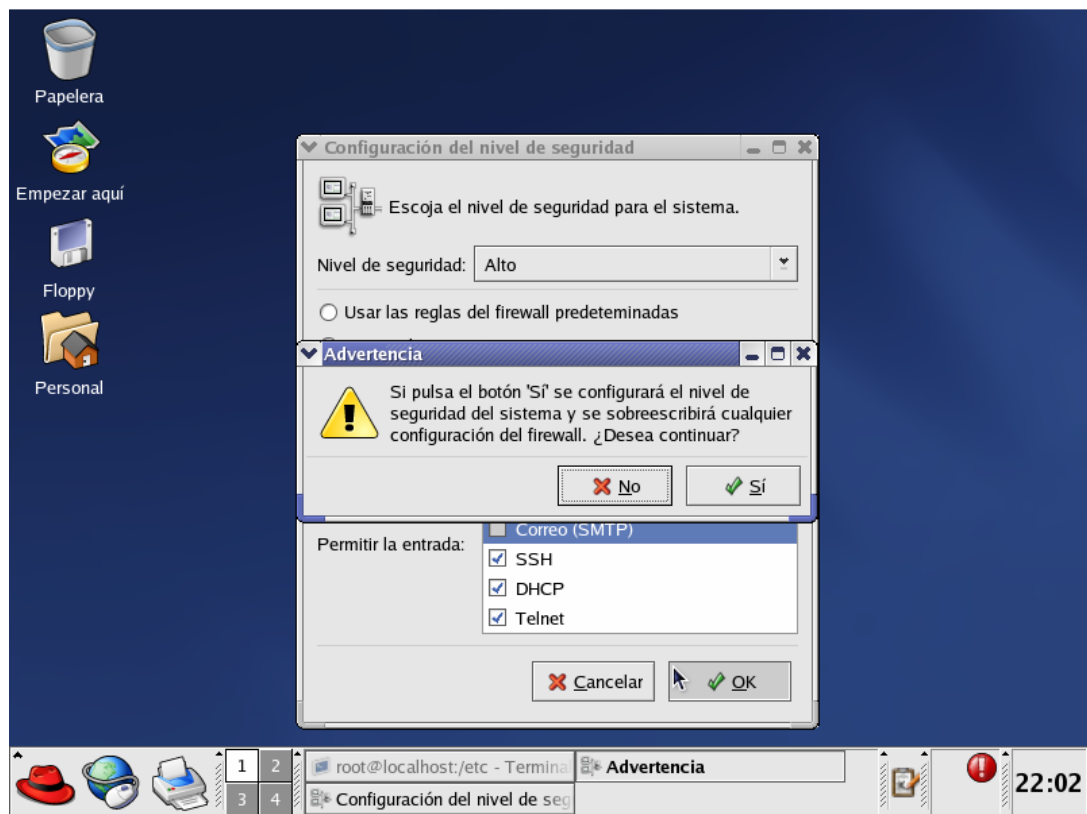
1. HABILITACIÓN DEL FIREWALL DEL SERVIDOR



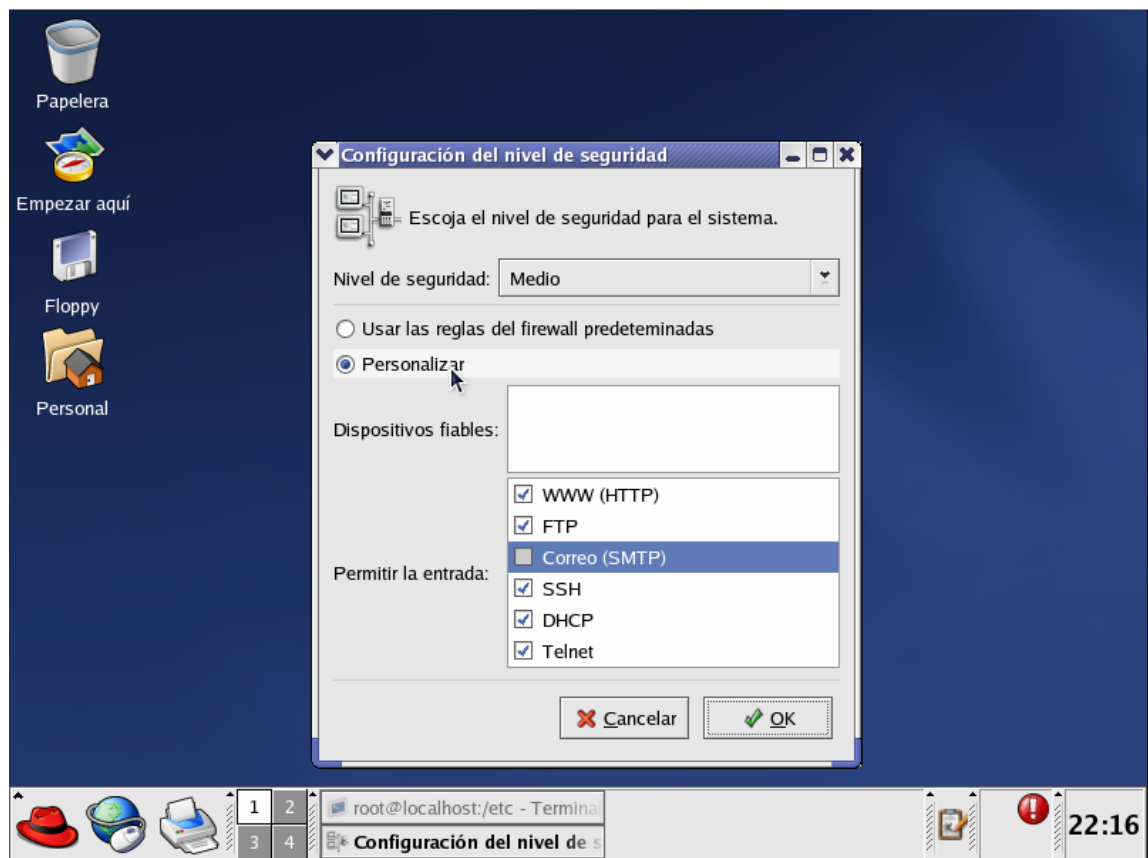
Para habilitar el firewall tendremos que ir a la barra de inicio del sistema, luego seleccionar la opción de configuraciones del sistema, después ir a nivel de seguridad.



Como siguiente paso asignaremos el grado de seguridad para nuestro firewall



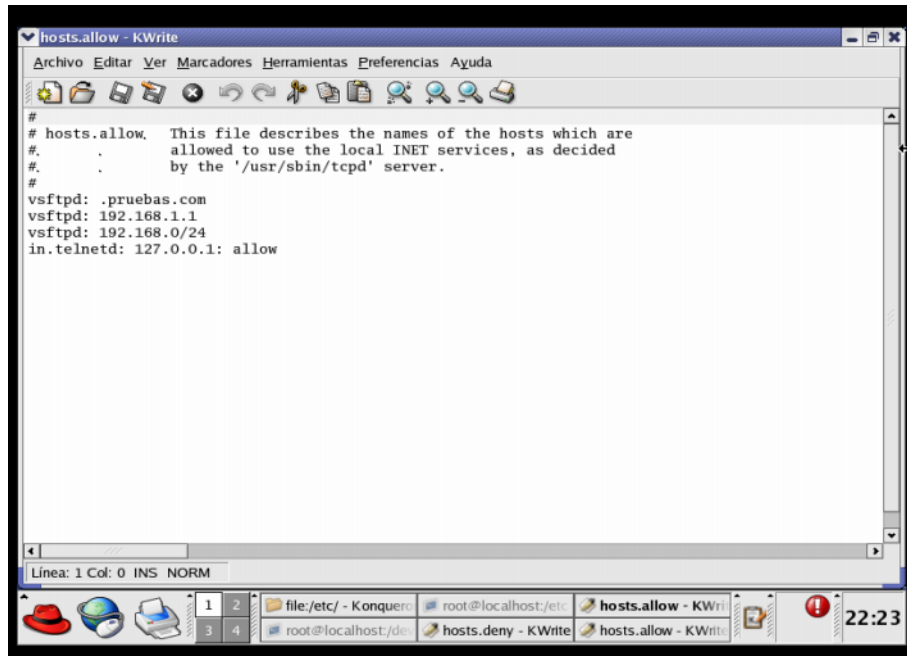
Si nuestra primera idea como administradores del sistema es ; poner el nivel más alto en el firewall. Se tiene que tomar en cuenta que esto dará como resultado una restricción importante en el sistema operativo y por consiguiente el administrador tendrá que tener un mayor conocimiento para poder manejar los resultados de esa configuración .



La opción mas viable es un control personalizado, en este caso seleccionaremos nivel intermedio con esto se lograra una selección de servicios que se estén utilizando ya que no todos los servicios serán necesario tenerlos con un nivel de seguridad alto.

Dado que no todos los servicios serán utilizados, ya que esto dependiendo de la función que dicho servidor este dando a la red .

2. CONFIGURACIONES DE RESTRICCIONES DE IP CON XINETD PARA CONEXIONES.

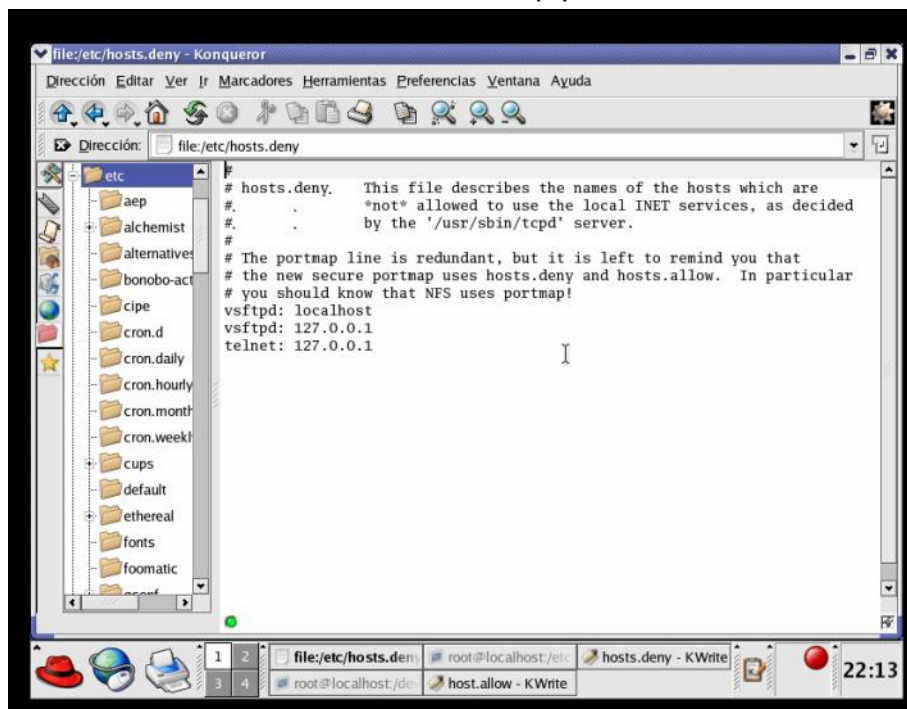


A screenshot of a KWrite window titled 'hosts.allow - KWrite'. The window shows the configuration for the 'hosts.allow' file. The text inside the window is as follows:

```
# hosts.allow This file describes the names of the hosts which are
#             allowed to use the local INET services, as decided
#             by the '/usr/sbin/tcpd' server.
#
vsftpd: .pruebas.com
vsftpd: 192.168.1.1
vsftpd: 192.168.0/24
in.telnetd: 127.0.0.1: allow
```

The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Marcadores', 'Herramientas', 'Preferencias', and 'Ayuda'. The status bar at the bottom shows 'Linea: 1 Col: 0 INS NORM' and a taskbar with various icons and a clock showing 22:23.

Servicio de xinetd “allow”. Acceso a ip para el Server

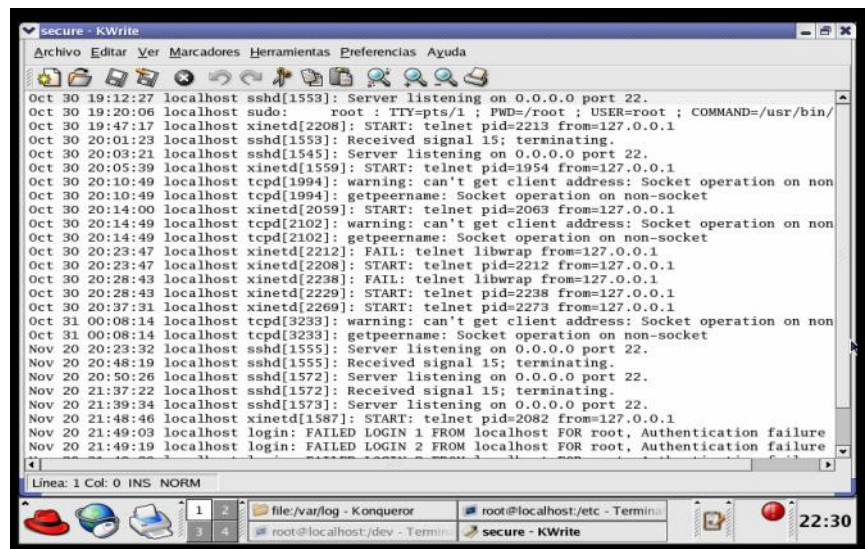


A screenshot of a KWrite window titled 'file:/etc/hosts.deny - Konqueror'. The window shows the configuration for the 'hosts.deny' file. The text inside the window is as follows:

```
# hosts.deny This file describes the names of the hosts which are
#             *not* allowed to use the local INET services, as decided
#             by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow. In particular
# you should know that NFS uses portmap!
vsftpd: localhost
vsftpd: 127.0.0.1
telnet: 127.0.0.1
```

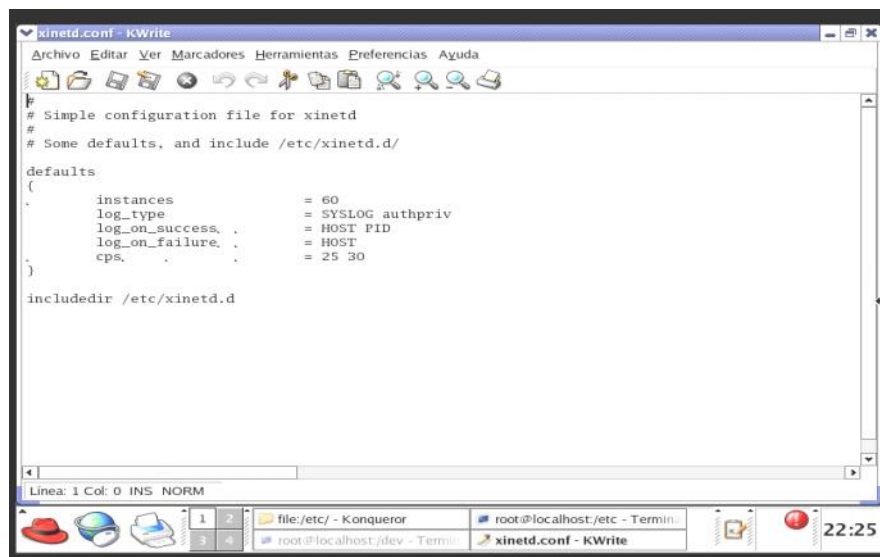
The window has a menu bar with 'Dirección', 'Editar', 'Ver', 'Ir', 'Marcadores', 'Herramientas', 'Preferencias', 'Ventana', and 'Ayuda'. The status bar at the bottom shows 'file:/etc/hosts.deny' and a taskbar with various icons and a clock showing 22:13.

Servicio de xinetd “deny”. Denegación de ip para entrar al Server



```
Oct 30 19:12:27 localhost sshd[1553]: Server listening on 0.0.0.0 port 22.
Oct 30 19:20:06 localhost sudo: root : TTY=pts/1 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/
Oct 30 19:47:17 localhost xinetd[2208]: START: telnet pid=2213 from=127.0.0.1
Oct 30 20:01:23 localhost sshd[1553]: Received signal 15; terminating.
Oct 30 20:03:21 localhost sshd[1545]: Server listening on 0.0.0.0 port 22.
Oct 30 20:05:39 localhost xinetd[1559]: START: telnet pid=1954 from=127.0.0.1
Oct 30 20:10:49 localhost tcpd[1994]: warning: can't get client address: Socket operation on non
Oct 30 20:10:49 localhost tcpd[1994]: getpeername: Socket operation on non-socket
Oct 30 20:14:00 localhost xinetd[2059]: START: telnet pid=2063 from=127.0.0.1
Oct 30 20:14:49 localhost tcpd[2102]: warning: can't get client address: Socket operation on non
Oct 30 20:14:49 localhost tcpd[2102]: getpeername: Socket operation on non-socket
Oct 30 20:23:47 localhost xinetd[2212]: FAIL: telnet libwrap from=127.0.0.1
Oct 30 20:23:47 localhost xinetd[2208]: START: telnet pid=2212 from=127.0.0.1
Oct 30 20:28:43 localhost xinetd[2238]: FAIL: telnet libwrap from=127.0.0.1
Oct 30 20:28:43 localhost xinetd[2229]: START: telnet pid=2238 from=127.0.0.1
Oct 30 20:37:31 localhost xinetd[2269]: START: telnet pid=2273 from=127.0.0.1
Oct 31 00:08:14 localhost tcpd[3233]: warning: can't get client address: Socket operation on non
Oct 31 00:08:14 localhost tcpd[3233]: getpeername: Socket operation on non-socket
Nov 20 20:23:32 localhost sshd[1555]: Server listening on 0.0.0.0 port 22.
Nov 20 20:48:19 localhost sshd[1555]: Received signal 15; terminating.
Nov 20 20:50:26 localhost sshd[1572]: Server listening on 0.0.0.0 port 22.
Nov 20 21:37:22 localhost sshd[1572]: Received signal 15; terminating.
Nov 20 21:39:34 localhost sshd[1573]: Server listening on 0.0.0.0 port 22.
Nov 20 21:48:46 localhost xinetd[1587]: START: telnet pid=2082 from=127.0.0.1
Nov 20 21:49:03 localhost login: FAILED LOGIN 1 FROM localhost FOR root, Authentication failure
Nov 20 21:49:19 localhost login: FAILED LOGIN 2 FROM localhost FOR root, Authentication failure
```

Servicios de aplicación de restricciones de xinetd



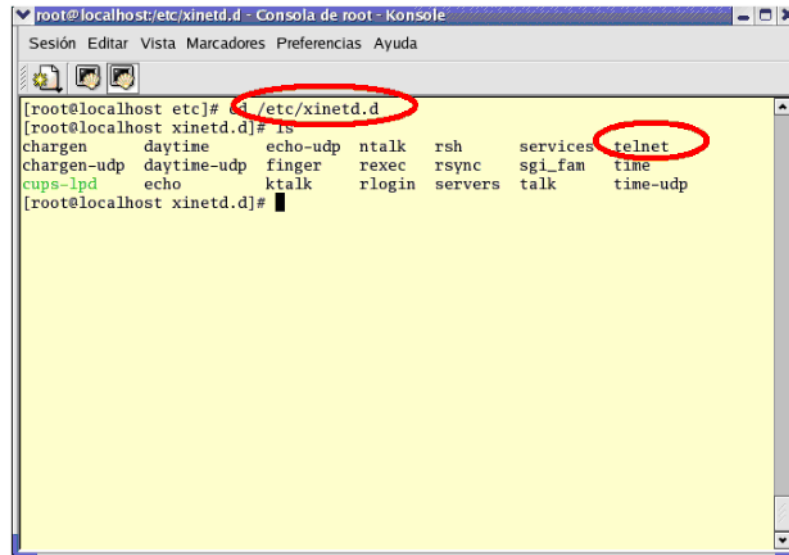
```
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
    instances                = 60
    log_type                  = SYSLOG authpriv
    log_on_success            = HOST PID
    log_on_failure            = HOST
    cps                       = 25 30
}

includedir /etc/xinetd.d
```

Bitácora de servicios levantados en el siste ma

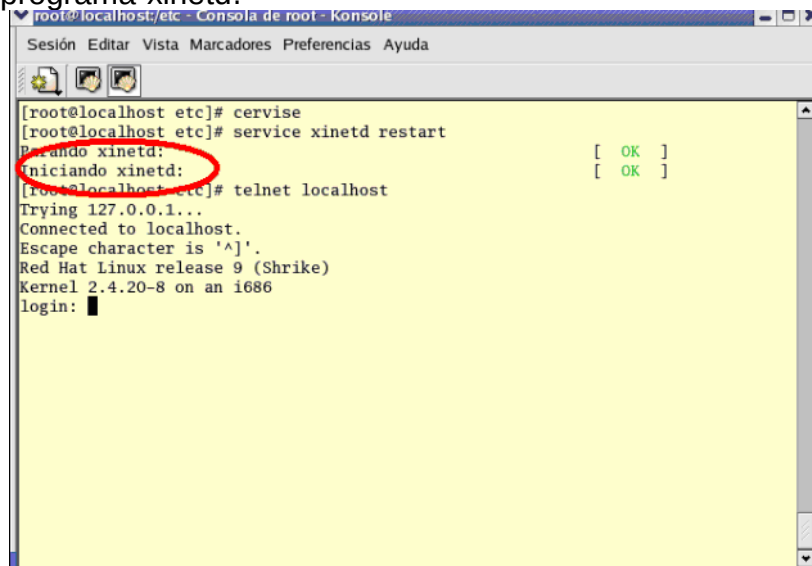
SERVICIOS DE XINETD



A terminal window titled "root@localhost:/etc/xinetd.d - Consola de root - Konsole". The command `cat /etc/xinetd.d` has been executed, displaying the contents of the `/etc/xinetd.d` directory. The output shows a list of service definitions, with `telnet` highlighted by a red circle. The terminal text is as follows:

```
[root@localhost etc]# cat /etc/xinetd.d
[root@localhost xinetd.d]# ls
chargen  daytime  echo-udp  ntalk    rsh       services  telnet
chargen-udp  daytime-udp  finger    rexec    rsync     sgi_fam   time
cups-lpd  echo      ktalk     rlogin   servers   talk      time-udp
[root@localhost xinetd.d]#
```

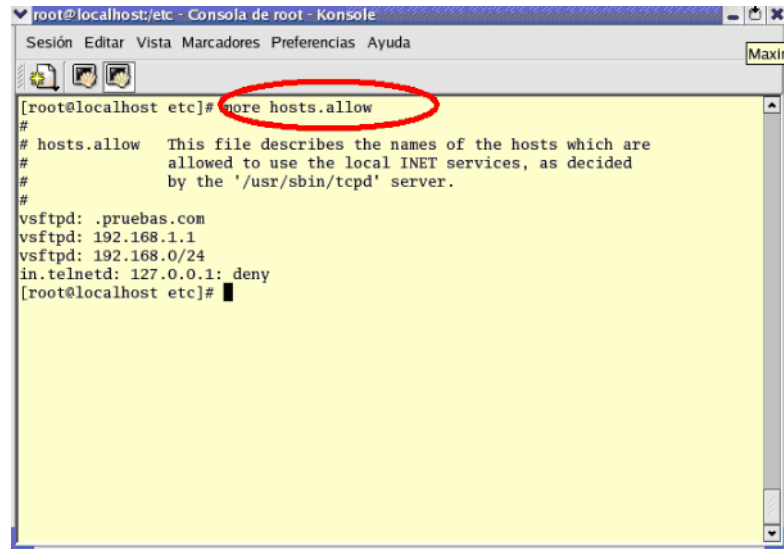
Para poder habilitar el servicio de Telnet al usuario **local host** es necesario iniciar el programa xinetd.



A terminal window titled "root@localhost/etc - Consola de root - Konsole". The commands `cervise`, `service xinetd restart`, and `telnet localhost` have been executed. The output shows the status of the service and the details of the telnet connection. The terminal text is as follows:

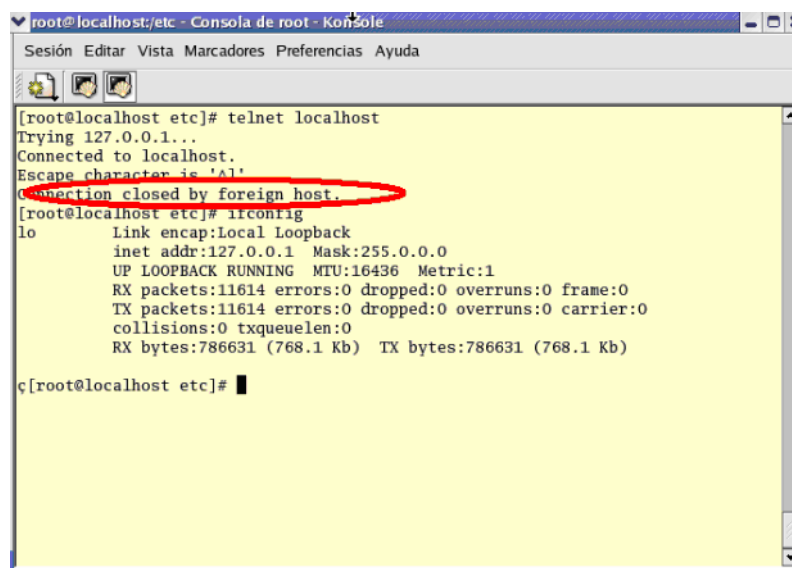
```
[root@localhost etc]# cervise
[root@localhost etc]# service xinetd restart
Parando xinetd: [ OK ]
Iniciando xinetd: [ OK ]
[root@localhost etc]# telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.
Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686
login:
```

Este servicio es para la asignación de conexión de un usuario por medio de Telnet con el servidor, el cual es necesario configurar antes de asignar el permiso correspondiente al usuario.



```
root@localhost/etc - Consola de root - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
[root@localhost etc]# more hosts.allow
#
# hosts.allow This file describes the names of the hosts which are
#             allowed to use the local INET services, as decided
#             by the '/usr/sbin/tcpd' server.
#
vsftpd: .pruebas.com
vsftpd: 192.168.1.1
vsftpd: 192.168.0/24
in.telnetd: 127.0.0.1: deny
[root@localhost etc]#
```

Ejemplo de asignación de habilitación de conexión de una IP usuario 127.0.0.1. al servidor por medio del Telnet utilizando el comando **allow** (permitir)



```
root@localhost/etc - Consola de root - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
[root@localhost etc]# telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^['
Connection closed by foreign host.
[root@localhost etc]# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:11614 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11614 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:786631 (768.1 Kb)  TX bytes:786631 (768.1 Kb)

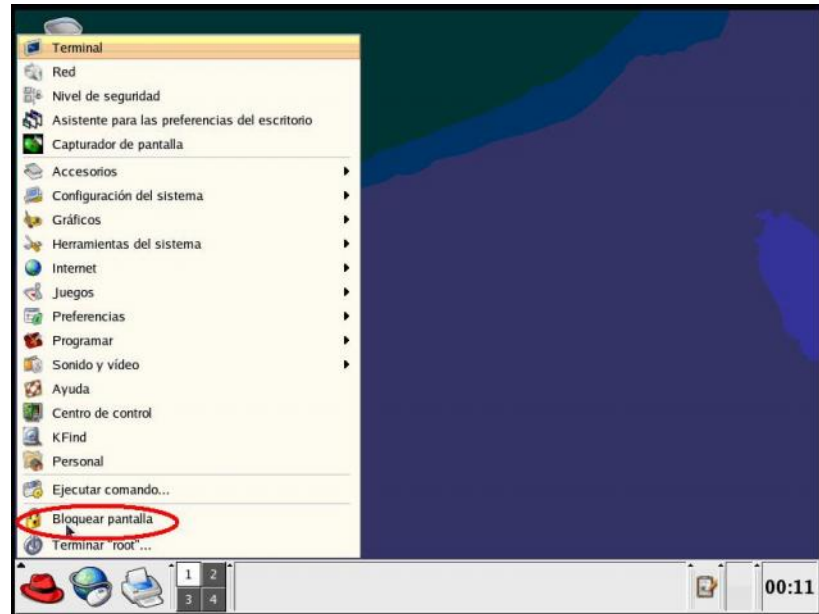
[root@localhost etc]#
```

Ejemplo de denegación de conexión servicio de Telnet a la IP usuario utilizando el comando **deny (denegar)**

[illegible]

Ejemplo de reasignación de permiso de conexión Telnet al usuario local host.

LOCK SCREEN



En la barra de menú de KDE se encuentra la opción de bloquear pantalla con la cual se habilita "LOCK SCREEN" para habilitar el bloqueo de la pantalla principal del sistema.



Cuando deseamos volver a trabajar en nuestro sistema, se tendrá que introducir la contraseña de usuario.

CONCLUSION

Actualmente no existe material educativo para capacitar a los estudiantes de la carrera de Técnico en Ingeniería de Redes Computacionales de la Universidad Tecnológica de El Salvador, en el ambiente de seguridad sobre el sistema operativo Linux red Hat 9.0. Es por ello que se presenta un manual técnico sobre configuración de estrategias de seguridad, reforzándolo además con material adicional en un CD en forma to acrobat, el cual posee material didáctico de fácil interpretación y permitirá que los estudiantes tengan acceso a este manual técnico.

Este manual técnico, puede utilizarse en materias que se imparten en la escuela de informática, tales como: Sistemas Operativos de Red, Administración de Redes, y cualquier otra materia que se pueda incluir o esté contemplada en el contenido temático relacionada a la seguridad de redes.

Como grupo de investigación se esta seguro; que el material didáctico entregado ayudara en gran medida a mejorar los niveles de seguridad a nivel de servidores de Internet en la plataforma de Red Hat 9.0 , ya que cuenta con información actualizada y de fácil comprensión para que cualquier estudiante con conocimientos básicos para que pueda interpretarlo y ponerlo en práctica.

RECOMENDACIONES

Como grupo de investigación que damos en mutuo acuerdo en presentar las siguientes recomendaciones:

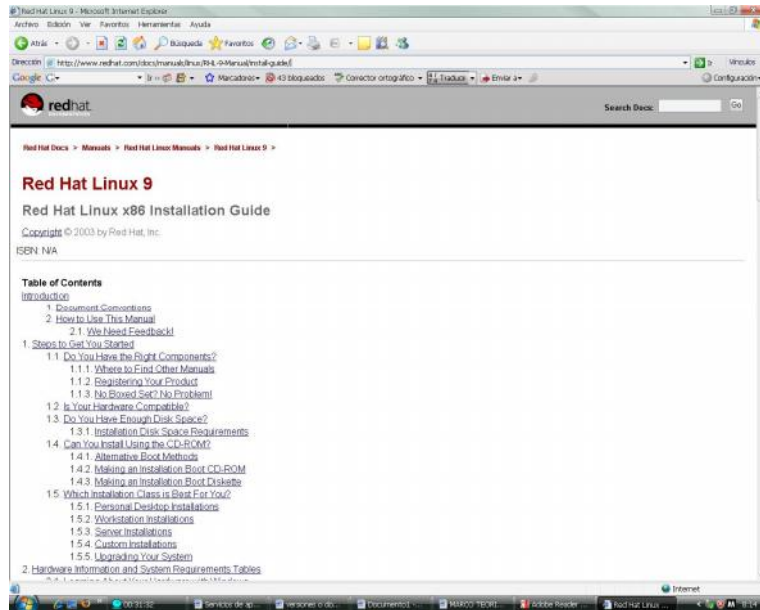
- o Tener una mayor diversidad de libros en la biblioteca que contenga información sobre seguridad en el sistema operativo Linux.
- o Mas apertura a la enseñanza del sistema operativo Linux, ya que; en el plan de estudio no se profundiza sobre el tema, y este ; en el futuro puede ser una herramienta en el ámbito laboral.

BIBLIOGRAFÍA

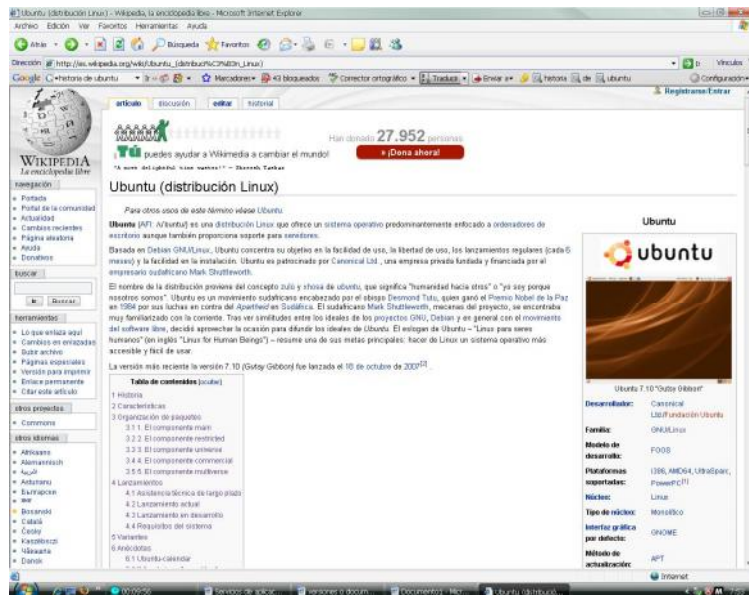
HACH, Brian & LEE, James. ***Hacker en Linux.*** 2^a Ed. Madrid, Mc Graw - Hill, 2003. P. 824

ANEXOS

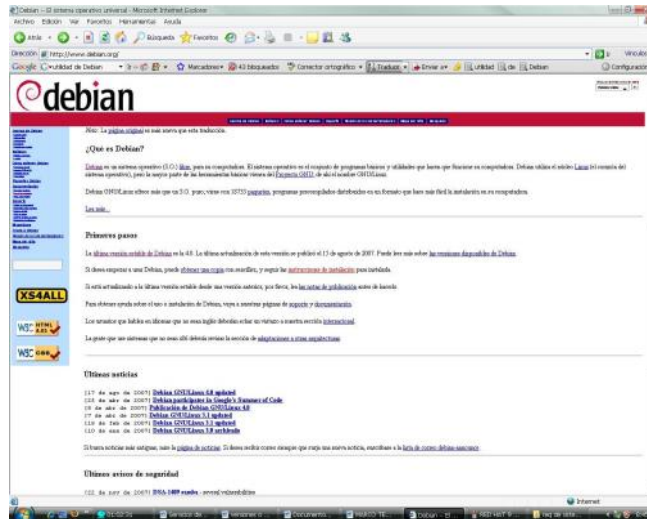
VERSIONES DE LINUX



<http://www.redhat.com/docs/manuals/linux/RHL-9-manual/install-guide/>

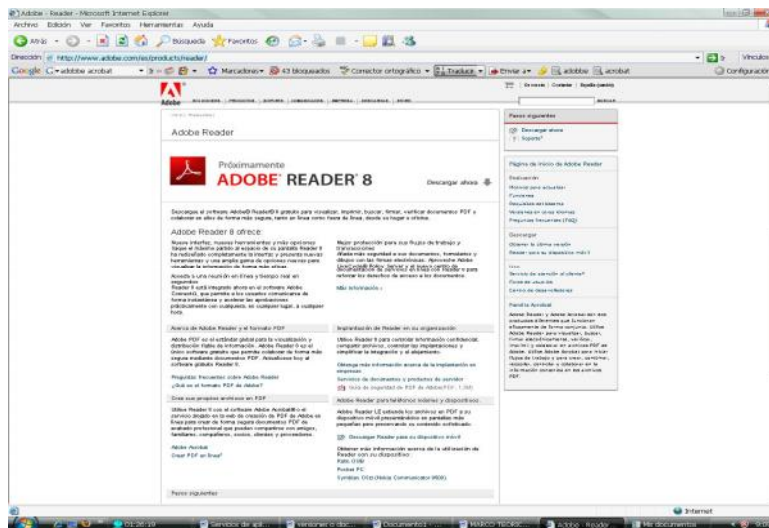


[http://es.wikipedia.org/wiki/Ubuntu_\(distribuci%C3%B3n_Linux\)](http://es.wikipedia.org/wiki/Ubuntu_(distribuci%C3%B3n_Linux))
www.ubuntu.com

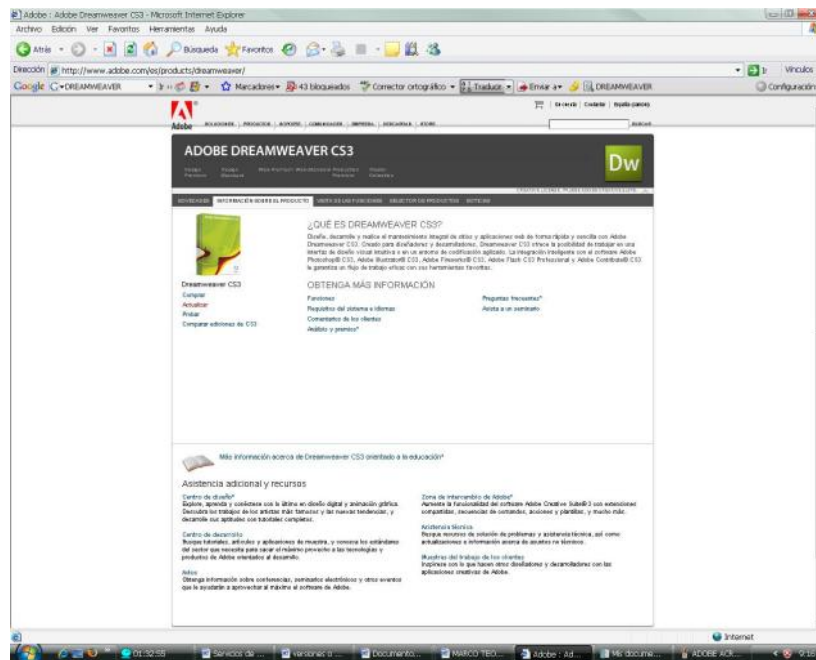


<http://www.debian.org/releases/stable/i386/ch03s04.html.es>

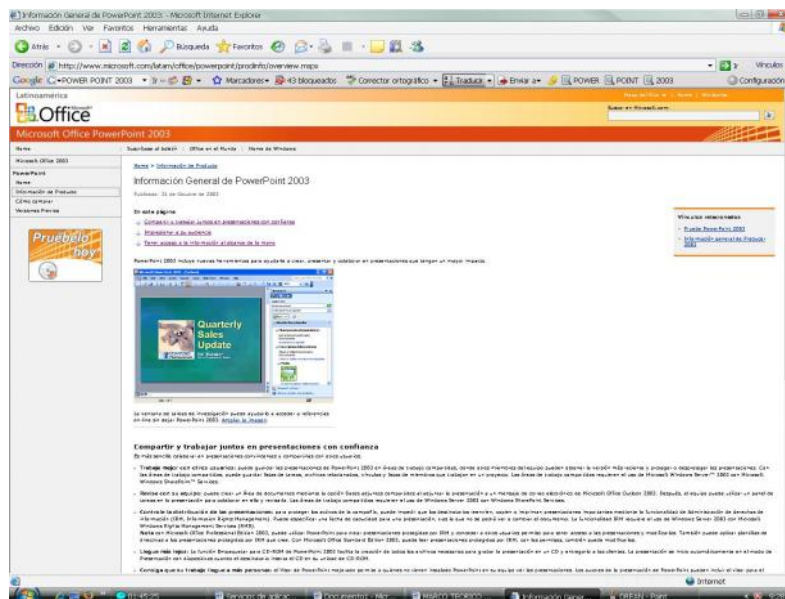
EDITORES DE TEXTO



<http://www.adobe.com/es/products/reader/>



<http://www.adobe.com/es/products/dreamweaver/>



<http://www.microsoft.com/latam/office/powerpoint/prodinfo/overview.msp>