41132

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGÓN

INGENIERÍA EN COMPUTACIÓN

"Propuesta de Integración de un Esquema de Seguridad en el Centro de Cómputo de una Entidad Gubernamental, Utilizando Herramientas de Libre Distribución"

TESIS

Que para obtener el Título de

INGENIERO EN COMPUTACIÓN

P r e s e n t a :

Rosendo Fabián Hernández

Asesor: M. en C. Jesús Díaz Barriga

México, D.F. **Junio 2004**

Agradecimientos

Esta tesis es un símbolo que representa la culminación de muchos años de estudio y aprendizaje, los cuales no hubieran sido posibles sin la bendición de Dios y sin el apoyo de tantas personas que fueron parte de este camino lleno de obstáculos y satisfacciones.

Agradezco a mis padres (Rosendo y Raquel) por su inmenso amor y por darme el mas preciado tesoro que se le puede otorgar a un hijo, el estudio. Por el apoyo incondicional que me dieron para convertir un sueño en realidad, por la confianza que han depositado en mi y sobretodo por enseñarme que la humildad es lo mas importante para salir adelante. Nunca podré pagarles con nada el sacrificio que hicieron por mi, los quiero mucho y esta tesis se las dedico a ustedes.

Este espacio es para dedicarle este trabajo a las 2 personas que han transformado mi vida, mi esposa Elizabeth, que día con día me hace entender el gran valor que tiene una mujer y agradecerle por haberme dado lo mejor que puede existir en la vida, un hijo (Axel). A ti Axel te dedico este esfuerzo y le doy gracias a Dios por quitar tus alas y dejar que viajaras hasta aquí para estar con nosotros y día con día regalarme tu risa, te amo bolita.

También quiero agradecer a mis hermanos que han sido ejemplos a seguir para poder cumplir mis metas. Jacobo, por enseñarme que rodearte de amistades demuestra lo que vales como persona. Arturo, eres la definición perfecta de humildad y sencillez, el hombre sensato y tranquilo digno de admiración. A ti Minerva, por tu ayuda incondicional que me has brindado siempre y mostrarme como se deben cumplir las metas. A mis cuñadas, Luisa y Catarina por apoyarme en los momentos buenos y malos.

Quiero dedicarles esta tesis a mis sobrinos, Jacobín, Moy, Ary, Arturín, Dany y las preciosas gemelas Lulú y Paty para que les sirva como fuente de inspiración en sus estudios y que sepan que al comenzar un camino hay que terminarlo.

Durante este largo trayecto de aprendizaje he conocido a muchos amigos, por lo que considero injusto no mencionar a alguno de ellos, por lo tanto, solo quiero agradecerles por todas las cosas que me han enseñado y los momentos inolvidables que me han hecho pasar a su lado.

No dejare de mencionar y agradecer a las personas que me asesoraron esta tesis.

Gracias, a todas las personas que han creído en mi......

Rosendo Fabián Hernández

3. Capítulo 3. Instalación y configuración de las herramientas de seguridad de libre distribución.

OBJETIVOS

Objetivo General

El objetivo general de esta tesis es diseñar esquemas de seguridad utilizando herramientas de libre distribución, que sirva de apoyo a todos los administradores de servidores Unix y que la consideren como una excelente opción de asesoría y a su vez difundan la cultura de seguridad en estos ambientes.

Objetivos Específicos

Investigar las herramientas de libre distribución existentes e identificar que cumplan con los requisitos necesarios para los esquemas de seguridad a implantar, pero dejando en claro que existen más herramientas (comerciales), no solo las ocupadas en esta tesis [1].

Recabar información del centro de cómputo, sobretodo de sus necesidades y deficiencias en cuanto a seguridad se refiere, para poder fortalecer y proteger la información importante que existe dentro de los servidores normales y críticos.

Obtener asesoría de personas con amplia experiencia en el campo de la seguridad que será de gran valor y aporte a esta tesis, ya que la retroalimentación en este campo es demasiado importante y provechosa.

Para esta tesis se conoce bien el ambiente en donde se implantarán los esquemas, pero de no ser así, se requiere de un estudio minucioso del lugar donde se implantará el esquema seleccionado, como servicios que usan, aplicaciones existentes, tipos de usuarios, horas laborales, nivel de importancia de cada servidor, impacto que tendrá en el performance y tiempo estimado de implantación (desde el diseño hasta el funcionamiento).

1 Utilizamos el término "software de libre distribución" a los programas de aplicación que están disponibles de manera gratuita y legal en internet. El término "software comercial" se refiere a los programas que venden las empresas dedicadas al desarrollo de software.

PREÁMBULO

Hace algunos años, el problema de la seguridad en cómputo surgió por la inquietud de entrar a sistemas prohibidos, ya sea por espionaje o por simple curiosidad, entre las historias más famosas se encuentran :

HISTORIA

Delito informático

Sucedió a finales de los 80's, en un centro de cómputo en donde se tenía un equipo IBM (línea 3400), que brindaba el servicio de pago de sueldos. El administrador del servidor se dió cuenta de que podía lograr la emisión de cheques de personas que ya se habían dado de baja y que ya no trabajaban en la empresa, sin embargo, las personas que llevaban la contabilidad se dieron cuenta 2 años después, que existían algunos errores y que algunos sueldos tenían sumados conceptos especiales, como horas extras y bonificaciones. Después de una auditoria descubrieron los fraudes de esta persona y fue sentenciada por la justicia. En este centro de cómputo, el sistema era demasiado seguro, dado que se cumplía aparentemente el orden, pero no fue así, por eso se recomienda no solo asegurar el hardware y software, si no también el personal de las áreas sistematizadas.

Juegos de Guerra

Durante 1983 se vivieron horas de extrema tensión en la central de la NORAD (North American Aerospace Defense Command) en la montaña de Cheyene, Colorado, USA, parecía que el mundo estaba en una guerra termonuclear. David Lightman, un estudiante de secundaria, había ingresado con su computadora y una conexión telefónica, al sistema de NORAD y comenzó un diseño táctico de guerra. Esta historia solamente es el argumento de la película de Juegos de Guerra, pero cabe mencionarla debido a que provocó una verdadera invasión de curiosos en los sistemas informáticos. El filme le abrió los ojos y la cabeza a miles de aficionados apasionados por la computación. A partir de entonces los mainframes de las grandes empresas, militares o del gobierno, así como muchas redes informáticas fueron blancos predilectos de los hackers, ansiosos por descubrir nuevos secretos.

Kevin Mitnick

Un internauta se caracteriza por usar un teléfono, un módem y una computadora para muchos fines diferentes, para Kevin Mitnick, el quehacer diario fue explorar y explotar computadoras ajenas y sistemas telefónicos. Mitnick pudo apropiarse de 20,000 números de tarjetas de crédito, sospechoso de robar software de más de media docena de fabricantes de celulares y controlar oficinas centrales de teléfonos en Manhattan.

La carrera de Mitnick inicia en 1980 cuando tenía apenas 16 años y rompió la seguridad del sistema administrativo de su colegio. Su bautizo como infractor de la ley fue en 1981,cuando entró a las oficinas de COSMOS (Computer System for Maintenance Operations) para controlar el registro de llamadas y obtener claves de seguridad y manuales del sistema. En 1982 entró ilegalmente a la computadora del North American Air Defense Command en Colorado. Un año mas tarde fue arrestado cuando era estudiante de a Universidad del Sur de California por entrar ilegalmente a ARPAnet y tratar de entrar a la computadora del pentágono. En 1987 fue acusado en Sta. Cruz de California de invadir el sistema de la compañía Microcorp Systems. Tres años después solicitó empleo en el Security Pacific Bank como encargado de la seguridad de la red del banco, pero fue rechazado por sus antecedentes penales, entonces Mitnick tomo represalias en contra del banco, falsificando un balance general, afortunadamente el administrador de la red detuvo el balance antes de que se efectuara y se perdieran millones de pesos. Ese mismo año inició el escándalo que lo lanzó a la fama, durante meses observó secretamente el correo electrónico de los miembros del departamento de seguridad de MCI communications y Digital equipment Corporation para conocer como estaban protegidas las computadoras y sus sistemas telefónicos. El personal de seguridad de Digital se dio cuenta inmediatamente del ataque y dieron aviso al FBI. Mitnick fue arrestado en 1988 por invadir el sistema de Digital Equipment.

Después de quedar en libertad, en 1994 se encontró con la computadora de Tsutomu Shimomura, la cual invadió en la navidad de ese año. Shimomura físico y experto en la seguridad en sistemas del San Diego Supercomputer Center, era un hacker "bueno", ya que cada que encontraba fallas en algún software, las reportaba a las autoridades. Mitnick le robó a Shimomura su correo electrónico, software para el control de teléfonos celulares y varias herramientas de seguridad. Shimomura lo detectó, se dio cuenta de que había invadido el sistema de Internex y lo comunicó inmediatamente al FBI, y juntos empezaron a rastrearlo inmediatamente. Shimomura envió un mensaje codificado a Netcom para notificarle de que el arresto de Mitnick se haría al día siguiente (16 de febrero). Llegada la hora el FBI procedió a arrestarlo. De regreso a su hotel Shimomura revisó su contestadora telefónica y escucho la voz de Mitnick que le había dejado varios mensajes, incluso dejo uno 8 horas después de su arresto, lo cual sigue siendo un misterio de como pudo realizarlo.

Entre las compañías afectadas por las actividades de Mitnick se encuentran : Motorola, Nokia, Fujitsu y Nec. Como se puede ver estas historias han marcado el rumbo de la seguridad en los sistemas Unix, en pro y en contra, ya que así como hay personas que con sus conocimientos intentan entrar a algún sistema, también hay personas que con esos mismos conocimientos se preocupan por proteger dichos sistemas. [2]

Esta tesis se ha elaborado esencialmente para los administradores de servidores Unix, sin el requisito de ser expertos y que les pueda servir como una guía importante en los aspectos de seguridad que quieran implantar en cualquier lugar. Esto no es una tarea sencilla, sin embargo, con los conocimientos básicos en lo que a Unix se refiere, es suficiente para idealizar un esquema e implantarlo de acuerdo a las necesidades de su centro de cómputo.

Todo lo mencionado solo es con el objetivo de ampliar el panorama de los administradores, en el área de seguridad, que día con día se va volviendo un peligro inminente al tener desprotegidos nuestros sistemas. Para poder elaborar esta tesis, se requirió de exhaustivas pruebas con cada una de las herramientas, las cuales se detallan en los capítulos 2 y 3.

2 Esta información fue tomada del libro "Takedown" de Tsutomu Shimomura y John Markoff (1997)dela editorial El Pais-Aguilar de 464 páginas. En el se relatan la búsqueda y captura de Kevin Mitnick.

AUDIENCIA

Esta tesis va dirigida a administradores de sistemas Unix para lograr un nivel alto de seguridad en sus equipos. La elaboración de ésta se debe al gran avance computacional que está desarrollándose en este país y, por lo tanto, mayor peligro en la información que se debe proteger en este ámbito.

1. CAPÍTULO 1 : ADMINISTRACIÓN Y SEGURIDAD EN SERVIDORES CON SISTEMA OPERATIVO UNIX.

1.1 INTRODUCCIÓN

El esquema cliente-servidor "es un modelo en el que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas se divide entre dos o más procesos", los principales componentes del esquema cliente-servidor son los ***clientes, los servidores y la infraestructura de comunicaciones.***

Los ***clientes*** se comunican con procesos que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta y realizar actividades de sincronización y de seguridad.

Los ***servidores*** proporcionan un servicio al cliente y devuelven los resultados. Existen procesos que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. También manejan servicios como administración de la red, mensajes, control y administración de la entrada al sistema, auditoria, recuperación y contabilidad. Usualmente en los servidores existe algún tipo de servicio de Base de Datos.

Para que los clientes y los servidores puedan comunicarse se requiere una ***infraestructura de comunicaciones,*** la cual proporciona los mecanismos básicos de direccionamiento y transporte. La mayoría de los sistemas cliente/servidor actuales se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer verificaciones de conexión. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

Ventajas del esquema cliente/servidor.

- Facilita la integración entre sistemas diferentes y, por lo tanto, compartir recursos.

- Se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes o de uso libre (freeware), lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.

- La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.

- El esquema cliente/servidor contribuye además a proporcionar a los diferentes departamentos de una empresa soluciones locales, pero permitiendo además la integración de la información relevante a nivel global.

- En este caso los mecanismos son distintos que en el caso de los sistemas centralizados. Por ejemplo, se deben hacer verificaciones en el cliente y en el servidor. También se puede recurrir a otras técnicas como el cifrado de la información

1.2 SISTEMA OPERATIVO UNIX

Este sistema operativo esta diseñado para : trabajar con el esquema cliente-servidor, brindar tiempos compartidos a las tareas que necesita ejecutar, controlar actividades y recursos de la computadora, correr múltiples procesos concurrentemente y soportar múltiples usuarios, lo cual hace más sencillo compartir la información entre ellos. El sistema operativo Unix fue diseñado con una arquitectura modular, por lo tanto, facilita su administración y permite instalar solo lo necesario.

1.3 ADMINISTRACIÓN DEL SISTEMA OPERATIVO UNIX

La administración de un sistema, en particular de Unix, se refiere a la instalación y mantenimiento del equipo en donde se instale el sistema operativo. Existen diferentes actividades importantes en la administración, como el mantenimiento del software y hardware, la configuración del hardware, la instalación del software, la reconfiguración del kernel, asignación de parámetros de red y cualquier cosa que requiera que el sistema trabaje y permanezca funcionando satisfactoriamente. El encargado de realizar todas estas funciones es el administrador del sistema (root), el cual tiene los privilegios para efectuar las tareas de administración, que normalmente no están disponibles para un usuario normal.

1.3.1 Actividades de un administrador

Un administrador es el encargado de controlar los recursos de los servidores, mantener la integridad del sistema, garantizar un funcionamiento eficiente y es el responsable de que siempre esté disponible el equipo que se está administrando. Otras actividades que también debe de realizar son : crear y borrar cuentas de usuarios, conectar o desconectar hardware, instalar y actualizar software, respaldo y recuperación de la información, monitoreo del rendimiento del sistema, arreglar desperfectos, instalación de parches, comunicación con otros sistemas y manejo de la seguridad del sistema.

Este último punto es de suma importancia ya que el administrador debe de implementar políticas de seguridad y periódicamente verificar que la seguridad de su sistema no ha sido violada. Para poder obtener un alto nivel de seguridad se requiere poner mucho énfasis en esta parte, ya que no solo se tiene que vigilar el equipo local sino todo su entorno (otros servidores, clientes y el tráfico en la red). A continuación se recomendarán algunas políticas a seguir e inmediatamente se comenzará a tocar el tema principal de esta tesis, el cual es la seguridad.

1.3.2 Políticas del administrador

Se deberán generar políticas enfocadas hacia las personas que vayan a utilizar el equipo que se esta administrando, entre las más importantes están las siguientes:

Políticas de software a utilizar, de usuarios, de clientes (software), de tiempo de uso, de cuotas, de comandos, de respaldo y de revisión de bitácoras.

Políticas de software a utilizar. Esta se refiere al software que se encuentra en el servidor como compiladores, aplicaciones gráficas, bases de datos, etc. Se deberá realizar un análisis del software necesario para cada usuario y solo se le darán los permisos de ejecución sobre éste.

Políticas de usuarios. Se establecen las características que deberá de tener la cuenta, como nombre de usuario, directorio $HOME, shell, UID y GID, además de definir si es un usuario que necesita la cuenta. Aquí también se le asigna una contraseña, éste es uno de los puntos mas importantes ya que es la manera que un equipo valida a un usuario, las políticas de la contraseña dependen de cada administrador, como la longitud de ésta (de preferencia mínimo 8 caracteres) , la fecha de vencimiento, que no sea fácil de adivinar y que sea personal e intransferible.

Políticas de clientes (software). Aquí se define que cliente usarán para conectarse al servidor, de preferencia debe de tener opciones de cifrado de información, para que la transferencia sea segura.

Políticas de tiempo de uso. Se podrá evaluar el uso del equipo, las horas en que se tendrá acceso a él y cuanto tiempo.

Políticas de cuotas. Se podrán asignar cuotas de espacio en disco y de uso de procesador.

Políticas de comandos. Se especificaría que comandos no estarán disponibles a cualquier usuario, solo los que requiera o sea los mas básicos.

Políticas de respaldo. Las horas que se emplearán para realizar los respaldos de sistema operativo y de las aplicaciones que estén instaladas, ya que algunas requieren que no se estén ocupando.

Políticas de revisión de bitácoras. Se revisarán las bitácoras de una manera constante y que arroje resultados de posibles fallas en tiempo real.

Estas son algunas políticas que se deben de considerar al momento de administrar un servidor con sistema operativo Unix, todas están enfocadas a la protección del sistema y que cuente con una buena seguridad y protección.

1.4 SEGURIDAD EN CÓMPUTO.

Un sistema es seguro si se comporta como el usuario y el administrador lo desean. Aunque se sabe que no se puede tener una seguridad total, si se podría llegar a tener un buen nivel de protección. Algunas cosas que se deben de considerar al momento de implantar seguridad son: computadoras, respaldos, datos, terminales, software y usuarios, este último es muy importante ya que la seguridad es responsabilidad de todos los que usan el sistema.

Un administrador debe de procurar brindarle a los usuarios :

Privacidad o confidencialidad. Esto se logra teniendo herramientas de autenticación, niveles de autorización y permisos.

Integridad. Asegurar que la información no será alterada y conservará su estado original

Disponibilidad. La información deberá estar disponible en todo momento y actualizada.

1.5 LA SEGURIDAD EN EL SISTEMA OPERATIVO UNIX.

Una de las funciones de un sistema operativo multiusuario y multitarea es permitir a usuarios y/o programas interactuar entre ellos. Cuando no se cuenta con protección alguna en cualquier programa o usuario, esto puede afectar a otros programas o a otros usuarios, ya que podrían borrar o modificar archivos importantes u ocasionar la baja del equipo.

El sistema operativo Unix fue diseñado con el objetivo de proporcionar un ambiente abierto. Dennis Ritchie, uno de los diseñadores originales de Unix, escribió lo siguiente: "El sistema operativo Unix no fue diseñado en un principio para ser seguro, fue diseñado con las características necesarias para ser servicial ".

Unix tiene un sofisticado sistema de seguridad que controla la manera en que los usuarios acceden a los archivos, el uso de los recursos del equipo y el contenido de los archivos de configuración. Estos mecanismos son inútiles cuando un sistema está mal configurado, descuidado o si tiene software con "bugs". Mucho de los huecos de seguridad encontrados en Unix son el resultado de esta clase de problemas mas que por el diseño del sistema.

A continuación se muestran algunas reglas básicas de seguridad :

- No poner archivos en nuestro sistema que probablemente sean de interés personal o sean confidenciales.
- Evitar tener huecos de seguridad en nuestro sistema. Esto lo podemos evitar leyendo boletines de seguridad, estar en listas de discusiones, actualización de software y aplicación de parches.
- Evitar servidores de ftp inseguros, cuentas con contraseñas débiles y cuentas grupales.

- Habilitar software que detecte inconsistencias en el sistema, sobre todo si este se conecta a Internet.

- Monitorear el sistema y atender cualquier anomalía encontrada. Si se encuentra algún problema, por menor que sea, deberá atenderse de manera inmediata

- Monitorear cualquier actividad inusual o rara, esto es, si en las bitácoras reporta cambios con alguna cuenta como : mas actividad, uso de la cuenta en horas anormales o en días anormales (fines de semana o vacaciones).

1.6 SEGURIDAD BÁSICA EN EL SISTEMA OPERATIVO UNIX

A continuación se mencionarán cuestiones básicas de seguridad que se deben implementar al estar administrando un sistema operativo Unix.

1.6.1 El archivo /etc/passwd

Este archivo contiene información de las cuentas del sistema, incluyendo el password (cifrado). Es la primer defensa de un sistema, ya que al momento de que se quiera establecer una conexión, este es de los primeros archivos que revisa, para poder validar y establecer dicha conexión, por lo tanto este archivo debe estar en constante revisión.

Uno de los puntos que tienen que verificarse es que ninguna cuenta, excepto la del super-usuario (root) debe de tener UID = 0. También se deben de cuidar los permisos de este archivo, ya que debe tener lectura para todos, pero solo de escritura para root: 644.

Unix permite que cada usuario tenga la facilidad de cambiar su password, pero esto es un gran inconveniente, ya que no se garantiza que el usuario asigne un password difícil de adivinar, por lo tanto se pone en riesgo la integridad del sistema. A continuación se darán algunas políticas que se deben de seguir al momento de escoger una contraseña:

- La contraseña deberá llevar como mínimo 8 caracteres.
- Se recomienda que lleve números, signos de puntuación, mayúsculas y minúsculas.
- No se deben escoger palabras sencillas de adivinar.
- Debe ser fácil de recordar para evitar anotarlo en algún lugar.
- Se recomienda modificarla por lo menos cada mes o cada 2 meses.

1.6.2 El archivo /etc/shadow

Debido a que el archivo /etc/passwd tiene permisos de lectura para todos, es muy peligroso, ya que por medio del uso de alguna herramienta (crack) se puede descifrar la contraseñan de cualquier usuario incluyendo root, es por eso, que es conveniente usar el archivo /etc/shadow. Este archivo es de acceso restringido, ya que solo root puede acceder a él por los permisos que tiene (400), aquí se guardan las contraseñas cifradas de los usuarios del sistema.

Cuando se ocupa el /etc/shadow, el archivo /etc/passwd tendrá una "x" en el campo del password del usuario y es lo único que varía cuando en un sistema existe el shadow.

Para generar este archivo se utiliza el comando pwconv y para deshabilitar el shadow se ocupa el comando pwunconv.

1.6.3 Permisos de los archivos y directorios

Unix brinda la facilidad de otorgar privilegios o permisos a archivos, esto nos garantiza un poco de seguridad, ya que con esto podemos escoger que grupo de usuarios tienen acceso a nuestros archivos. A continuación se mencionan algunos puntos importantes que siempre se deberán de tomar en cuenta al momento de asignar permisos en archivos y directorios.

Archivos

- Los permisos de un archivo dependen del dueño, del grupo y del directorio en el que se encuentre. Esto es, si un archivo no tiene permisos de escritura para cualquier usuario, éste se podrá borrar si el directorio en donde esta tiene los permisos necesarios para realizar esta operación.
- Los archivos aparecen con el nombre de su dueño, si éste es borrado del sistema (y quitado del archivo /etc/passwd), estos aparecerán con el UID del usuario.
- Si el usuario se quita permisos a nivel de dueño y conserva estos a nivel de grupo, no podrá hacer nada con el archivo, ya que tienen mayor prioridad los permisos a nivel de dueño.
- El usuario root tiene acceso a todos los archivos del sistema.
- Cuando se tiene permisos de lectura sobre un archivo, aún sin tener de ejecución, se puede hacer una copia del mismo y cambiarle los permisos, pero el archivo cambia de dueño (al que hizo la copia).
- Para los shell-scripts es necesario que estos tengan los permisos de lectura y el de ejecución, ya que el intérprete va leyendo el código y lo va ejecutando.
- Cuando se le cambian los permisos a una liga dura, esto afecta a lo demás.
- Los permisos en una liga suave, no afectan los permisos del archivo. Cuando se intenta cambiar el permiso de la liga, realmente se modifica el permiso del archivo.
- La opción –R permite cambiar los permisos recursivamente en directorios.

Los permisos que existen se muestran a continuación:

r	**lectura**
w	**escritura**
x	**ejecución**
s	**SUID o SGID**
t	**sticky bit**

Ahora se muestra los permisos con números:

0400	**Lectura para el dueño**
0200	**Escritura para el dueño**
0100	**Ejecución para el dueño**
0040	**Lectura para el grupo**

0020	Escritura para el grupo
0010	Ejecución para el grupo
0004	Lectura para otros

Directorios

- Cuando se tiene permiso de lectura en un directorio, pero sin tener el de ejecución se puede realizar un listado corto sobre el (ls), pero no un listado largo (ls –al).

- Es necesario tener permisos de ejecución en un directorio para poder entrar a él.

- Teniendo permisos de ejecución pero no de lectura en un directorio no se pueden listar los archivos, pero se pueden acceder y ejecutar los archivos individualmente, obviamente conociendo los nombres del archivo.

- Cuando no se tiene el permiso de ejecución en un directorio no se pueden acceder a sus archivos, aún siendo el dueño de este.

1.6.4 Archivos con permisos SUID, SGID y sticky bit

En Unix existen permisos especiales para otorgar privilegios especiales, estos son:

4000	SUID
2000	SGID
1000	sticky bit

El bit de SUID o setuid se activa otorgándole a un archivo permisos de ejecución al dueño y de SUID (4000), entonces en las primeras 3 letras en lugar de la **x** aparecerá una **s** o una **S** (esta aparece si no se le ha puesto el permiso de ejecución arriba mencionado, pero no tendría efecto el SUID). Cuando se activa el SUID, indica que todos los que ejecuten ese archivo tendrán los privilegios de quien lo creó, mientras lo estén ejecutando. Si root crea un archivo y le pone SUID, todo usuario que ejecute este archivo, va a obtener privilegios de administrador.

El bit de SGID se aplica de la misma manera que el SUID pero esto se realiza a nivel de grupo, no a nivel de dueño. Cuando un usuario ejecuta un programa con SGID tendrá los privilegios del grupo al que pertenece. EL SGID se asigna sumándole 2000 a los permisos del archivo y asignando permisos de ejecución al grupo y en lugar de la **x** aparecerá una **s** o una **S** (esta aparece si no se le ha puesto el permiso de ejecución arriba mencionado, pero no tendría efecto el SGID). Si se aplica el SGID a un directorio este afectará a archivos y subdirectorios que estén dentro de el, ya que como grupo al mismo que el directorio con el SGID.

Los bits SUID y SGID otorgan una gran flexibilidad y fácil manejo de archivos y directorios, pero a su vez son un gran problema de seguridad, ya que los usuarios buscan este tipo de archivos para obtener más privilegios, sin embargo, hay muchos archivos del sistema que necesitan estos permisos forzosamente y el más común es el /bin/passwd, el cual debe de tener estos privilegios, sin embargo hay otros que no es necesario que los tengan, tal es el caso de /usr/sbin/ping, entonces la tarea de administrador del sistema es de minimizar la existencia de archivos con SUID y SGID, modificándoles los permisos a estos. Se recomienda revisar continuamente la existencia de archivos con estos permisos.

También existe el sticky bit, el cual se activa sumándole 1000 a los permisos del archivo o directorio y con los permisos de ejecución a todos los usuarios y en lugar de la **x** aparecerá una **t** o una **T** (esta aparece si no se le ha puesto el permiso de ejecución arriba mencionado, pero no tendría efecto el sticky bit).

Este permiso indica que aparentemente todos los usuarios pueden crear y eliminar archivos dentro un directorio, solo el dueño del archivo y el administrador podrán manipular el archivo guardado dentro de este directorio. Este permiso se usa principalmente en directorios en donde todos los usuarios del sistema puedan escribir y solo puedan borrar su información, un ejemplo típico es el directorio /tmp, pero se debe de tener cuidado, por que en algunas plataformas al tirar el equipo, desaparece esta información.

Algunas recomendaciones para los programas con SUID:

- No realizar shell-scripts con SUID, ya que regularmente en estos existen variables de ambiente del dueño o se invoca a otros archivos del mismo usuario.
- Usar rutas absolutas dentro de los programas, no usar rutas relativas, por que se podría ejecutar este programa desde otro directorio y afectar a otros archivos.
- No realizar programas con SUID y con permiso de lectura para todos, solo de ejecución.
- Evitar los caracteres de escape en los programas con SUID, esto es, usando el argumento trap y poniendo los números correspondientes de escape al shell.

1.6.5 Los archivos /etc/hosts.equiv y .rhosts

En el archivo /etc/hosts.equiv se pueden poner los servidores (en forma de lista) que se consideran confiables, esto es, que al utilizar los comandos remotos (rlogin, rsh o rcp) no será necesario proporcionar alguna contraseña. Por ejemplo si en el archivo /etc/hosts.equiv del servidor llamado server1 existe esta línea:

```
server2
```

En el equipo server2 podemos teclear lo siguiente:

```
server2 % rlogin server1
```

Y nos conectará al equipo server1 sin solicitarnos contraseña alguna, siempre y cuando en los 2 equipos exista el mismo login de usuario. Si en este archivo existiera un signo de "+" en lugar del nombre de otro equipo, indica que nuestro equipo confía en todos los equipos, esto sería demasiado arriesgado.

El manejo del archivo .rhosts es muy similar al de /etc/hosts.equiv, solo que éste puede existir en el directorio HOME de cualquier usuario, y permite la confiabilidad entre servidores pero a nivel de usuario. Al igual que el archivo anterior a éste se le asignan los equipos confiables y el usuario podrá conectarse desde estos equipos sin que se le solicite su contraseña, pero al igual debe tener el mismo login de usuario.

Si root tuviera este archivo (.rhosts) y le asignará la línea siguiente:

```
+        +
```

indicaría que confía en cualquier equipo y en cualquier usuario, éste es el peor caso, puesto que cualquier usuario con un mismo login de cualquier máquina se podrá conectar sin ninguna restricción.

La manera más sencilla de evitar el uso de estos archivos, es deshabilitando el uso de los comandos remotos, que veremos un poco más adelante

1.6.6 Listas de control de acceso: ACL's

Las listas de control de acceso brindan un nivel de seguridad mayor en los archivos, ya que asignan permisos a ciertos usuarios o grupos, es decir, no es necesario que 2 usuarios pertenezcan al mismo grupo para asignarles permisos de acceso a cierto archivo. Para poder ver si una archivo tiene ACL's se utiliza el comando getfacl y el nombre del archivo:

% getfacl archivo

el cual nos desplegará una lista con la ruta completa del archivo, el dueño del archivo, el grupo del archivo y los permisos que tiene éste.

Con el comando setfacl le asignamos permisos de listas de control de acceso a un archivo, se puede usar la opción –m seguida de los permisos que deseamos poner separados por comas, también se puede utilizar la opción –s para cambiar la ACL completa y para borrar los permisos con la opción –d.

opción tipo: UID | GID: permisos

tipo : indica a que se le aplicará los permisos (user, group, mask)
UID ó login : es el usuario al que se le asignarán los permisos
GID ó nombre del grupo : es el grupo al que se le asignarán los permisos
Permisos : Aquí se ponen los permisos que se otorgarán, ya sea con letras o de manera octal.

Otro campo que se debe modificar antes de asignar permisos con ACL's es el mask del respectivo archivo, por ejemplo,

% setfacl –m mask:r-x archivo

Aquí se le está asignando una máscara de permisos al archivo (similar al umask), en este caso la máscara es de lectura y ejecución.

El manejo de ACL's es muy útil para el administrador del sistema ya que puede tener un mayor control sobre archivos específicos, sin embargo, hay que tener cuidado con los archivos que estén manejando ACL's, es recomendable verificar continuamente los archivos que las están manejando, y la manera de darnos cuenta es por medio de un signo "+" que aparece al final de la lista de permisos de un archivo:

```
-rwxr--r--+
```

1.6.7 Servicios de red

/etc/hosts

Este archivo es muy útil cuando no se tiene un servidor de DNS, ya que funciona de manera muy similar, debido a que éste tiene una lista con el nombre de los equipos relacionados con su dirección IP, por ejemplo:

```
132.248.51.4        ultra2        ultra2.ciencias.unam.mx
```

esto es, se puede hacer referencia a este equipo por cualquiera de las 3 maneras.

/etc/services

Este es uno de los archivos de configuración de los servicios de red locales, su formato es el siguiente:

```
Servicio      puerto/protocolo      alias
telnet        23/tcp                telnet
```

Este archivo se ocupa para obtener el número de puerto por el que debe escuchar y al que se le envían las peticiones desde éste, pero se puede utilizar cualquier otro puerto para cualquier servicio. En este archivo no se pueden deshabilitar servicios, solo sirve para obtener puertos default a partir de los nombres de servicio.

/etc/inetd.conf

Uno de los archivos más importantes, si no es que el más importante es el archivo /etc/inetd.conf, éste es el encargado de ofrecer servicios de red a los clientes, por medio del demonio inetd.

Cuando el demonio inetd recibe una petición, se va a este archivo a ejecutar la línea correspondiente, la cual tiene el siguiente formato:

```
Servicio socket  protocolo    status    UID    programa servidor
telnet   stream  tcp          nowait    root   /usr/sbin/in.telnetd
```

donde:

servicio : es el nombre del servicio solicitado
socket : es el tipo de socket asociado a la conexión, por ejemplo para TCP regularmente se usan streams y para UDP se utilizan dgram o datagramas
protocolo : debe ser un protocolo reconocido en el archivo /etc/protocols, generalmente se usa TCP y UDP
status : se utiliza generalmente para sockets de tipo dgram (para stream se asigna nowait), si se pueden atender varias peticiones a la vez se pondrá nowait, en caso contrario, que solo pueda atender una a la vez se pone wait, que indica que tiene que esperar a que se libere para poder ocuparlo.

Algunos Unix permiten especificar junto a wait o nowait, un número máximo de peticiones a su servicio durante un minuto, de tal forma que si este número es excedido, inetd deja de ofrecer este servicio durante cierto tiempo. Por ejemplo si se quiere tener solo 30 peticiones al servicio finger durante un minuto, se pondría:

```
finger stream  tcp    nowait.30  nobody  /usr/sbin/in.fingerd    in.fingerd
```

1.6.8 Algunos servicios de red peligrosos

finger

Éste es uno de los servicios que ocasionan más problemas a un servidor, por la información tan detallada que brinda acerca de los usuarios que están conectados a un servidor. Los datos que nos proporciona son los siguientes:

```
Login          Name                 TTY     Idle     When            Where
admin  Operador del servidor        pts/1    55    Wed 18:46     10.150.61.148
```

login : Nombre del usuario
name: Detalles del usuario (tomado del campo GECOS del /etc/passwd)
TTY : Terminal que le asignó el sistema operativo
Idle : Tiempo de espera
When: Fecha y hora de conexión
Where : Dirección IP de donde se conectó

Como se nota, esta información es muy completa para alguien que quiera realizar un ataque, ya que por lo menos sabe el login de un usuario del sistema y su dirección IP.

Servicios rlogin, rsh y rcp

Para poder utilizar estos servicios se necesita de un cliente y un servidor, del lado del cliente se ejecutan los comandos remotos (rlogin, rsh, rcp) y del lado del servidor se requiere que estén corriendo los respectivos demonios (rlogind, rshd, rexecd). El rlogin se ocupa de manera muy similar al telnet, el rsh se utiliza para ejecutar comandos en un servidor sin entrar a este y el rcp es similar al ftp.

El objetivo de estos servicios remotos es evitarle al usuario final el uso de las contraseñas, pero obviamente implica un gran problema de seguridad, sin embargo su uso se basa en la confiabilidad entre usuarios y equipos, ya que un equipo puede ser confiable si su nombre se encuentra en el archivo /etc/hosts.equiv o si esta en el archivo .rhosts el cual estará en el HOME del usuario correspondiente. En el caso de que se encuentre en el /etc/hosts.equiv, requerirá que en el cliente y servidor exista el usuario con el mismo login, sin importar las contraseñas. En el caso del archivo .rhosts cualquier usuario del equipo remoto se podrá conectar al servidor, pero con el login del usuario que tiene el .rhosts bajo su $HOME.

Ejemplos del uso de estos servicios:

rlogin

```
serv1 :~# rlogin -l admin serv2
serv2 :~#
```

en este caso no se pide contraseña y el usuario admin se conecta al servidor 2

rsh

```
serv1:~# rsh -l admin  serv2 id
uid=1000(admin) gid=100(users) groups=100(users)
```

se ejecuta el comando id sin ninguna restricción

rcp

```
serv1:~# rcp  prueba.txt  admin@serv2:/tmp/
```

copia el archivo prueba.txt en el directorio /tmp/ del sistema remoto, bajo la identidad del usuario admin.

Es notorio que la confianza entre los equipos Unix es muy cómoda para los usuarios. Sin embargo al mismo tiempo se confía plenamente en los demás equipos, que a su vez probablemente estén confiando en otros, y si estos se ven comprometidos en su seguridad, también la del servidor que confía en ellos. Se recomienda habilitar estos servicios solo en el caso de que los equipos sean administrados por la misma persona. También es conveniente verificar que no exista el archivo .rhosts en los directorios $HOME de los usuarios, una manera de evitar esto es crear en estos directorios el archivo .rhosts con permisos 000, para que ningún usuario pueda modificarlo.

Es necesario deshabilitar estos servicios tanto para uso local (/etc/services) como para uso remoto (/etc/inetd.conf), solo comentando con un # al principio de la línea respectiva.

En el archivo /etc/services:

```
#exec        512/tcp
#login       513/tcp
#shell       514/tcp
```

En el archivo /etc/inetd.conf

```
#shell  stream  tcp   nowait  root   /usr/sbin/in.rshd      in.rshd
#login  stream  tcp   nowait  root   /usr/sbin/in.rlogind   in.rlogind
#exec   stream  tcp   nowait  root   /usr/sbin/in.rexecd    in.rexecd
```

talk y write

Estas utilerías están enfocadas a establecer comunicación entre 2 usuarios, por lo tanto, es recomendable que el sistema no tenga permitido usar estos servicios de mensajería instantánea que ofrece Unix, ya que la información viaja en claro por la red, además de que las terminales que se ocupan no son del todo seguras (como se verá más adelante), y la información que se pueda suministrar puede ser confidencial y peligrosa.

Para deshabilitar el talk solo es necesario comentar líneas en los siguientes archivos:

/etc/services

```
#talk     517/udp
```

y **/etc/inetd.conf**

```
#talk  dgram  udp   wait   root   /usr/sbin/in.talkd     in.talkd
```

Para no permitir el uso del comando write, basta solo con cambiar los permisos de este comando, que originalmente los tiene en:

```
-r-xr-sr-x
```

y dejárselos para que solo lo pueda ejecutar root. Otra opción para deshabilitarlo seria con el comando mesg y la opción n:

```
%mesg n
```

con esto no permitirá recibir mensajes el usuario.

daytime

Este servicio está asociado al puerto 13, y al momento de recibir una conexión a este puerto, el sistema mandará la información de fecha y hora del sistema.

```
serv1 % telnet serv2 13
wed Oct 10 19:12:46 CDT 2001
se perdió conexión con el host
serv1 %
```

Aparentemente esto no es un riesgo de seguridad ya que no brinda información sobresaliente del sistema (solo la ubicación geográfica del equipo), sin embargo, es un servicio que no es de utilidad y que no requerimos tenerlo disponible. Otro servicio muy similar al daytime es el time (puerto 37), solo que este se ocupa para sincronizar estaciones cliente con el reloj de un servidor utilizando los comandos netdate o rdate:

```
serv1# rdate –s serv2
```

Con este comando el serv1 se sincroniza con la hora del serv2, aquí serv1 actúa como cliente y serv2 como servidor.

netstat

Similar al systat, solo que este ofrece información del estado de la red. Utiliza el puerto 15 para mostrar conexiones activas en un servidor, por ejemplo:

```
serv1 # telnet serv2 netstat
Active internet connections
Proto    recv-Q send-Q local Address      foreign address     state
tcp           0    0     serv2:netstat      serv1:4555     ESTABLISHED
serv1#
```

Se recomienda deshabilitar este servicio ya que esta información puede ser de mucha utilidad para una persona malintencionada, por que está mostrando información como: nombre del servidor, servicios de red que presta y tipos de conexión que establecen los usuarios.

chargen

Este ocupa el puerto 19 y es un generador de caracteres que regularmente se ocupa para ver el estado de las conexiones de la red, al momento de solicitar este servicio solo se ve una secuencia de caracteres ASCII, los cuales se repiten indefinidamente, esto puede ocasionar negaciones de servicio, ya sea en el cliente o en el servidor.

tftp (trivial ftp)

Este programa se ocupa para transferir archivos y su uso esta basado en el protocolo UDP, por lo tanto, no es muy confiable, ya que no está brindando ningún tipo de seguridad. Su principal uso es permitir a estaciones de trabajo inicializarse a través de la red (bajar el kernel de un servidor) o poder transmitir cualquier archivo a cualquier directorio sin pedir usuario ni contraseña. Es recomendable revisar este servicio o de preferencia deshabilitarlo del archivo /etc/inetd.conf

```
#tftp  dgram  udp   wait  root  /usr/sbin/in.tftpd   in.tftpd -s /tftpboot
```

FTP (File Transfer Protocol)

Como su nombre lo indica es un protocolo de transferencia de archivos, pensado en ofrecer una rápida transferencia pero sin seguridad, ya que toda la información que se envía viaja en claro. Es recomendable que en estos casos se utilice software que permita enviar información cifrada, como secure shell, el cual se verá a detalle en capítulos posteriores.

Algunas recomendaciones al usar el FTP son: el superusuario (root) no necesita de este servicio ya que solo trabaja en consola, otros usuarios que no necesitan son uucp, daemon y bin. Para que todos estos usuarios no tengan permisos de conexión vía ftp, se crea un archivo llamado /etc/ftpusers con el siguiente formato:

```
# vi /etc/ftpusers

root
bin
uucp
daemon
```

estos usuarios no tendrán el servicio de FTP.

FTP Anónimo

Como se vió en el tema anterior el servicio de FTP es peligroso, pero cuando se instala un servidor de FTP anónimo es mucho más. A continuación se muestra la configuración de este servidor de manera correcta. Primero es necesario crear el usuario ftp teniendo como dueño de su directorio $HOME a root y el grupo debe pertenecerle a uno llamado ftp, con esto se tiene que los permisos de usuarios solo son para el administrador y los de grupo para los usuarios anónimos.

Después, dentro del $HOME de ftp se crea un árbol de directorios mínimos (/etc y /bin, que el dueño sea root y con permisos 111) para poder trabajar, esto es, usando el comando chroot, el cual permite a los usuarios anónimos ver el directorio raiz (/) de su conexión como si fuera tal, aunque realmente es el siguiente: ($HOMEftp/).

Debajo de /etc/ estarán los archivos passwd y group pero sin las contraseñas, solo como referencia. En el directorio /bin, estarán copias de los comandos que deseemos que ejecuten los usuarios que se conecten como anónimos. Otro directorio que se debe de crear es $HOME/pub, el cual tendrá directorios de deposito de información que se compartirá, sus permisos serán 555. También se creará un directorio llamado

$HOME/incoming el cual nos servirá para que los usuarios puedan dejar su información en este directorio, aunque puede ser contraproducente, ya que nos podrían sobrescribir algún archivo o programa, para evitar esto se requieren los siguientes permisos en este directorio: 1733, esto evitará que sobrescriban dichos archivos.

Otra medida de seguridad es poner este directorio en una partición separada de las demás para evitar que alguien intente llenarlo, además poner cuotas, que se detallará más adelante.

Telnet

Este servicio ocupa el puerto 23, y es el que nos permite conectarnos de manera remota a un servidor estableciendo una terminal virtual, solo que sin la seguridad de estar físicamente enfrente del servidor. Este no ocupa ningún tipo de cifrado, por lo tanto, toda la información que se envíe a través de aquí, viajará en claro por la red (lo más peligroso seria el login y el password). Se recomienda usar otro tipo de software para lograr la conexión de manera remota, el más común y de libre distribución es el Secure Shell, que se detallará en el siguiente capítulo.

Otros problemas que existen con el telnet son los siguientes :

1) Se puede utilizar para saber que otros puertos están abiertos:
   # telnet IP_server pto
2) Para visualizar la versión de kernel
3) Para saber en que plataforma estamos trabajando.
4) Que se permita la conexión de root por medio de este comando

Se recomienda deshabilitar el desplegado de la versión del sistema operativo que muestra cuando se hace una petición por telnet.

En HP-UX se modifica la línea de telnet en el inetd.conf, agregándole la opción –b, como se muestra a continuación:

```
telnet    stream tcp nowait root /usr/lbin/telnetd  telnetd -b
```

En Solaris es necesario modificar el siguiente archivo /etc/default/telnetd, originalmente tiene la línea BANNER sin ninguna opción, hay que dejarla de la siguiente manera:

```
BANNER=""
```

NIS (Network Information Service)

La filosofía de NIS es tener una "base de datos" distribuida, para que diferentes equipos puedan compartir archivos de configuración como : /etc/passwd, /etc/group, /etc/hosts y otros. Trabaja mediante el esquema maestro-esclavo o cliente-servidor, en donde los clientes usan las bases de datos que se encuentran en el servidor (maestro) como si estuvieran de manera local, a continuación se mencionarán algunos aspectos importantes cuando se éste utilizando NIS.

Al utilizar NIS se requiere una línea especial en el archivo /etc/passwd (de los clientes), la cual comienza con un signo de +,

```
root:x:0:1:Super-User:/:/sbin/sh
+:*:0:0:::
```

y le indica a los programas que hacen uso de este archivo que el resto del archivo lo busquen en el servidor NIS. Esta línea no deberá de ponerse en el servidor, por que en este no tendría ningún significado, entonces el signo de + lo tomaria como otro usuario cualquiera.

NFS (Network File System)

NFS permite compartir archivos o filesystems a través de la red usando el esquema cliente-servidor. Un cliente NFS puede montar un disco de un servidor simulando que físicamente está de manera local, además de poder leer y modificar el contenido de los archivos que se encuentren en ste, y lo mas peligroso es que se puede montar sin que se tenga una cuenta en el servidor NFS, lo que implica muchos problemas de seguridad.

Las comunicaciones entre el cliente y el servidor NFS se basan en RPC (Remote Procedure Call), el cual permite a los programas correr rutinas en un sistema aunque sean ejecutadas en otro. También usa el protocolo UDP para tener una mayor velocidad en la transmisión, aunque se tiene menor confiabilidad, debido al orden de entrega de los paquetes o pérdida de estos, aunque NFS solicita al servidor una confirmación de cada comando RPC ejecutado y eéste le devuelve un código que indica si el comando se realizó exitosamente. En el momento que el cliente no recibe confirmación, entonces UDP pierde el comando original de RPC o la confirmación de RPC. En el primer caso no hay problema porque el servidor se da cuenta desde la primera vez que se transmitió, pero en el segundo caso el servidor recibe el mismo comando dos veces; para algunos comandos (ls) esto no importa, sin embargo otros (mkdir) no podrán ser ejecutados dos veces consecutivas.

Cuando el Filesystem se monta con la opción soft, habrá un timeout en una retransmisión, pero si es montado con la opción hard, el cliente seguirá enviando solicitudes hasta que reciba la confirmación. Un servidor NFS está diseñado como connectionless y stateless, el primero significa que el servidor no registra cada cliente que ha montado algun filesystem y stateless indica que la información que el cliente necesita para montar un filesystem se encuentra en el cliente, y la ventaja de ambos es que a pesar de que se pierda la conexión con el servidor, por cualquier motivo, los clientes podrán seguir usando el filesystem por que no hay que reestablecer la conexión, de manera inmediata.

1.6.9 Terminales seguras

Como todos los servidores Unix tienen una cuenta de root (super-usuario), es muy común que traten de conectarse de manera remota utilizándola para conectarse al servidor. Para evitarlo se recomienda restringir el acceso de cualquier terminal remota y permitir solo conectarse por la consola (de manera local). Lo anterior se realiza en el archivo /etc/default/login, modificando la línea de CONSOLE quedando de la siguiente manera:

```
CONSOLE=/dev/console
```

O en otros sistemas, en el archivo /etc/securetty se agrega lo siguiente:

```
console
```

Con esta restricción no se podrán conectar como root usando el comando telnet pero existe otra manera de intentar adquirir los perfiles del usuario root o de algún otro y es con el comando su.

1.6.10 su (substitute user).

Este comando nos permite (una vez establecida una sesión), cambiarnos de usuario, lo mas usual al usuario root. Si no es usado con argumento alguno, nos pide el password de root (por default), si se usa con la opción – se requerirá el nombre de algún usuario (incluyendo root). Se recomienda al usar telnet, conectarse como un usuario (que no sea root), y después switchearse al usuario root.

En muchos sistemas para poder usar el comando **su** es necesario pertenecer al grupo "wheel". Este comando tiene algunos problemas de registro en bitácoras, en la realización de respaldos y en restricciones de uso. Una solución a estos problemas es la utilería llamada sudo, la cual se verá a detalle en capítulos posteriores.

1.6.11 Cuotas de recursos del servidor a nivel de usuario

Como medida preventiva de agotamiento de espacio en disco o de un filesystem, es conveniente llevar un control de este recurso usado por todos los usuarios de un sistema, usando cuotas a nivel de filesystem. Estas nos permiten asignar limites, que se especifican por 2 números: uno "suave" que se conoce como soft limit, en el que, si un usuario rebasa este valor, el sistema le manda una notificación de violación de cuota y el "duro" (hard limit) que determina el límite absoluto de uso, el cual no podrá ser rebasado.

Para habilitar las cuotas se tiene que realizar lo siguiente:

1) Primero se tiene que modificar el archivo /etc/vfstab, en la línea que corresponda al filesystem que tendrá las cuotas :

/dev/dsk/c0t0d0s6 /dev/rdsk/c0t0d0s6 /export/home ufs 2 yes rq

donde :

/dev/dsk/c0t0d0s6 : es el dispositivo a montar
/dev/rdsk/c0t0d0s6 : es el dispositivo "crudo" del anterior
/export/home : punto de montaje
ufs : Tipo de filesystem
2 : Tipo de chequeo que se le da al filesystem
yes o no : Si se quiere que se monte al momento de bootear
rq : Opciones de montaje, donde la q indica que se ocuparán cuotas.

2) Una vez realizado esto, se tiene que crear un archivo que se llame quotas (el dueño será root y tendrá permiso 600), en el nivel más alto del flesystem que tendrá las cuotas, en este ejemplo sería en :

#touch quotas /export/home

3) Se levanta el sistema de cuotas con el comando : quota on
4) Con el comando edquota y el nombre del usuarios se abre el editor default, para poner los limites (soft y hard).

#edquota usuario

el formato desplegado es así:

```
FS /export/home kbytes (soft=8000 hard=10000) inodes (soft=0 hard=0)
```

5) Se modifican los correspondientes a Kbytes, en este ejemplo queda el soft de 8 MB y el hard de 10 MB

No deberá olvidarse de que si la cuota está en cero, es por qué no hay cuotas y que "hard" debe ser mayor que "soft".

1.6.12 Tablas de cron (crontab)

El cron es un demonio que permite programar tareas para que se ejecuten de manera automática en el momento indicado. Existe un directorio donde se guardan los trabajos programados, normalmente están en : /var/spool/cron/crontabs.

Se puede limitar el uso de este comando en un archivo que se llama cron.deny, aquí se le indicará que usuarios "no" podrán ejecutar el cron, también se puede crear el archivo cron.allow, que su funcionalidad es exactamente lo contrario, o sea, los usuarios que si pueden ejecutar el cron.

Algunas opciones de este comando son :

-l Despliega los trabajos que se programaron.
-r Con esta opción root puede eliminar trabajos pendientes de cualquier usuario.
-e Para editar el archivo en donde se programaron las tareas a ejecutar.

Estas tareas programadas son también un punto que se debe de revisar constantemente, para evitar que se ejecuten tareas innecesarias y sobretodo en horas "pico" de un servidor. Se recomienda que se restrinja el uso del cron para que solo root lo pueda usar, pero no siempre se puede lograr esto, entonces será mejor tener una revisión constante a estas tablas y verificar que es lo que se tiene programado.

Una manera de explotar la funcionalidad del cron, es a través de los permisos que tenga el directorio /var/spool/cron/crontabs, que es donde se alojan los trabajos pendientes. Este directorio puede tener permisos de escritura para todo mundo, entonces cualquier usuario podría modificar el archivo de cron y programar las tareas que desee, obviamente como root. Otro punto que se debe de cuidar es el contenido del archivo de cron, esto es, dentro de este archivo se pueden ejecutar algunos scripts o programas que tengan permisos de escritura y ejecución para todos, lo que permitiria la modificación del contenido de estos o cambiarlos por otros archivos con el mismo nombre.

1.6.13 Seguridad en el núcleo del sistema operativo (kernel)

El núcleo o kernel es la parte más importante del sistema operativo Unix, de hecho el kernel es considerado como el sistema operativo en sí, ya que utiliza todas las instrucciones del procesador, direcciona toda la memoria, accede directamente al hardware, por lo que un error en la programación o configuración del núcleo puede ser peligroso para el sistema, esto suele ocurrir en los diferentes tipos de Unix que permiten la modificación o inserción de módulos en el kernel, tal es el caso de linux, solaris o freeBSD.

1.6.14 Software y parches

Cuando se tiene un servidor en producción, es necesario instalar software de aplicación para que todos o algunos usuarios puedan desempeñar sus labores cotidianas, algunos ejemplos serían: utilerías de compresión de información, browsers de navegación por internet, procesadores de texto, graficadores, compiladores, etc. con cada uno se debe de tener cuidado para ver que usuarios pueden acceder a cada uno de ellos, sin embargo, en el que se debe de poner atención especial y un buen control es en cualquier tipo de compilador, ya que muchos programas que se distribuyen a través de internet, requieren de uno (los mas usados C y gcc) para poder actuar. Se recomienda restringir su uso (de ser posible), de lo contrario se deberá llevar un monitoreo constante de su uso y verificar si no se están compilando programas que puedan afectar el desempeño del servidor o peor aún, que sea un programa que pueda poner en riesgo la seguridad del servidor.

Parches y/o actualizaciones de vulnerabilidades.

Para cada versión nueva que sale de cualquier sistema operativo o software de aplicación, es común que se le encuentre algún problema en su integridad (vulnerabilidades) o en su desempeño, lo cual afecta al rendimiento o a la seguridad del servidor. Para resolver esto, después de que aparece el producto en el mercado, se distribuyen "parches" [1] que solucionan los problemas que se van detectando, entonces es muy importante mantenerse al tanto de la aparición de estos programas accediendo a las páginas en Internet de los diferentes sistemas operativos o suscribiéndose a listas encargadas de informar (de manera inmediata) de la aparición de un problema y la recomendación de la resolución de éste, ya sea a través de modificación de archivos y permisos o a través de la aplicación de un parche.

La manera de instalar los parches varía en cada sistema operativo, y se debe de tener mucho cuidado en la aplicación de alguno, por que como su nombre lo indica "parcha" (actualiza) el software, esto quiere decir que modifica el programa en sí, en el caso del sistema operativo, realiza cambio de archivos importantes o en muchas ocasiones modifica el "kernel" y esto indica que se tiene que dar de baja el equipo y volver a iniciarlo, en un ambiente en producción esto puede ser muy peligroso, por lo tanto, se recomienda:

1. Realizar un respaldo del sistema a "parchar"
2. Hacerlo en horas que no sean críticas
3. Leer el archivo de ayuda del parche, el cual indica que modificaciones realiza
4. Ejecutar los comandos de aplicación de parches de manera adecuada
5. No aplicar parches innecesarios, por que pueden desestabilizar el sistema
6. Después de instalar el parche, realizar pruebas de funcionalidad del software que se actualizó, sobre todo si fue el sistema operativo.

1.6.15 Errores en los programas (Buffer overflows)

Los buffer overflows ocurren cuando un programa o proceso intenta almacenar más datos de los que soporta un buffer (área temporal de almacenamiento de datos). Puesto que los buffers son creados para contener una cantidad finita de datos, la información extra en estos puede ocasionar un desbordamiento en los buffers contiguos, corrompiendo o sobrescribiendo los datos válidos que se tienen.

---

[1] Los parches son mejoras que se le hacen al sistema operativo despues de su liberacion, ya que regularmente tiene bugs que no fueron detectados a tiempo

Aunque esto puede ocurrir accidentalmente debido a errores de programación, los buffer overflows son una causa común de ataque de seguridad en la integridad de los datos. En ataques de buffer overflows, los datos extras pueden contener códigos diseñados para realizar acciones especificas, enviando nuevas instrucciones al equipo atacado, que puede dañar archivos de usuario, cambiar datos o dejar al descubierto información confidencial.

La manera de resolver este tipo de problemas que se presentan regularmente en programas de aplicación o de sistema operativo, es a través de los parches que se distribuyen, como se mencionó en el tema anterior.

1.7 AUDITORÍA DEL SISTEMA

Todas las actividades realizadas en un sistema Unix pueden ser monitoreadas aunque no a gran detalle, para eso se cuenta con otro tipo de herramienta (TCP-Wrappers), que se detallará mas adelante. Esta facilidad de recabar información tiene ventajas para la seguridad del sistema, ya que se podrá detectar un intento de ataque o usos indebidos de los recursos del sistema, sin embargo, existen desventajas, ya que la gran cantidad de información puede generar grandes volúmenes de datos.

Los archivos log en Unix son simples archivos de texto (generalmente) por lo que se pueden visualizar con cualquier editor de texto, esto facilita al administrador para programar shell-scripts que comprueben los informes generados automáticamente, con ordenes como awk, grep o sed, pero el hecho de que sean de texto también facilita a un atacante el ocultar ciertos registros modificando dichos archivos, por eso no se debe de confiar al 100% en lo que digan los informes de auditoria.

En los sistemas con system V el process accounting se puede iniciar ejecutando el archivo /usr/lib/acct/startup, y para ver las bitácoras se utiliza el comando acctcom, y para BSD los equivalentes a estas ódenes accton y lastcomm. El process acounting es un historial de los comandos que se vayan ejecutando en el servidor.

1.7.1 El demonio syslogd

Éste se lanza de manera automática al arrancar el sistema operativo, y se encarga de guardar informes sobre el funcionamiento del servidor, este recibe mensajes de diferentes partes del sistema (del núcleo, de los programas, de las conexiones, etc) y los almacena en diferentes archivos que están definidos en el archivo /etc/syslog.conf, en donde se especifican las reglas de almacenamiento.

Ejemplo del archivo /etc/syslog.conf :

```
#ident "@(#)syslog.conf     1.5     99/02/03 SMI"   /* SunOS 5.0 */
#
# Copyright (c) 1991-1999 by Sun Microsystems, Inc.
# All rights reserved.
#
# syslog configuration file.
#
# This file is processed by m4 so be careful to quote (`') names
# that match m4 reserved words. Also, within ifdef's, arguments
# containing commas must be quoted.
#
*.err;kern.notice;auth.notice                  /dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit       /var/adm/messages

*.alert;kern.err;daemon.err                    operator
```

```
*.alert                    root

*.emerg                     *

# if a non-loghost machine chooses to have authentication messages
# sent to the loghost machine, un-comment out the following line:
#auth.notice             ifdef(`LOGHOST', /var/log/authlog, @loghost)

mail.debug               ifdef(`LOGHOST', /var/log/syslog, @loghost)
#user.info               /usr/local/adm/
#
# non-loghost machines will use the following lines to cause "user"
# log messages to be logged locally.
#
ifdef(`LOGHOST', ,
user.err                   /dev/sysmsg
user.err                   /var/adm/messages
user.alert                 `root, operator'
user.emerg                  *
)
local0.info                /usr/local/adm/
```

Como se puede ver cada regla tiene 2 campos : uno de selección y uno de acción. El campo de selección está formado a su vez de 2 partes: una del servicio que envía el mensaje y otra de su prioridad, separadas por un punto, la parte del servicio contiene alguna de las siguientes palabras clave : auth, auth-priv, cron, daemon, kern, lpr, mail, mark, news, security (equivalente a auth), syslog, user, uucp, y local0 hasta local7. Esta parte especifica el subsistema que ha generado el mensaje. La prioridad está compuesta de uno de los siguientes términos, en orden ascendente : debug, info, notice, warning ó warn, err ó error, crit, alert, emerg y panic. La prioridad define la importancia del mensaje almacenado. Además de estos términos, syslogd emplea los siguientes caracteres especiales:

 # (gato)
Este indica que si se pone al principio de una línea, entonces ésta se tomará como comentario nada más

' * ' (asterisco)
Se usa como comodín para todas las prioridades y servicios

mail.   /var/adm/mail*

guarda todos los mensajes del servicio mail en /var/adm/mail

' ' (blanco, espacio nulo)
Indica que no hay prioridad definida para el servicio de línea almacenada

' , ' (coma)
Con este caracter es posible especificar múltiples servicios con el mismo patrón de prioridad en una misma línea

mail,news.=info   /var/adm/info

#guarda todos los mensajes mail.info y news.info en /var/adm/info

' ; ' (punto y coma)
Se pueden dirigir los mensajes de varios servicios y prioridades a un mismo destino, separándolos por este carácter

**.=info;*.=notice   /var/log/messages*

Guardar los mensajes de prioridad "info" y "notice" en el archivo /var/log/messages

'=' (igual)
Con este carácter se almacenan los mensajes con la prioridad exacta especificada y no incluyendo las superiores:

```
*.=crit                                   /var/adm/critical
```

Guardar todos los mensajes críticos en /var/adm/critical

'!' (exclamación)
Anteponer el campo de prioridad con un signo de exclamación sirve para ignorar todas las prioridades, teniendo la posibilidad de escoger entre la especificada (!=prioridad) y la especificada más todas las superiores (!prioridad). Cuando se usan conjuntamente los caracteres `=' y `!', el signo de exclamación `!' debe preceder obligatoriamente al signo igual `=', de esta forma: !=.

```
kern.info;kern.!err                       /var/adm/kernel-info
mail.*;mail.!=info                        /var/adm/mail
```

Guardar mensajes del kernel de prioridad info, pero no de prioridad err y superiores
Guardar mensajes de mail excepto los de prioridad info

Por su parte, el campo de acción describe el destino de los mensajes, que puede ser :

Un fichero plano
Si se precede cada entrada con el signo menos, '-', se omite la sincronización del archivo (vaciado del buffer de memoria a disco). Aunque puede ocurrir que se pierda información si el sistema se cae justo después de un intento de escritura en el archivo, utilizando este signo se puede conseguir una mejora importante en la velocidad, especialmente si estamos ejecutando programas que mandan muchos mensajes al demonio syslogd.

```
*.=crit                                   /var/adm/critical
```

Guardamos todos los mensajes de prioridad crítica en "critical"

Una terminal (la consola)
Existe la posibilidad de enviar los mensajes a terminales; se puede tener una de las terminales virtuales que muchos sistemas Unix ofrecen en su consola:

```
*.*                                       /dev/tty5
kern.crit                                 /dev/console
```

Enviamos todos los mensajes a tty5 y todos los mensajes críticos del núcleo a consola

Una máquina remota
Se pueden enviar los mensajes del sistema a otra máquina, esto es útil si tenemos una máquina segura, y nos servirá para detectar usuarios 'ocultos' en nuestro sistema:

```
*.warn                                    /usr/adm/syslog
*.*                                       @maquina2
```

Enviamos los mensajes de prioridad warning y superiores al archivo "syslog" y todos los mensajes (incluidos los anteriores) a la máquina "máquina2"

Usuarios del sistema
Se especifica la lista de usuarios que deben recibir un tipo de mensaje, simplemente escribiendo su login, separados por comas:

.alert   root, toni

Enviar los mensajes con la prioridad "alert" a root y toni

Todos los usuarios que estén conectados
Los errores con una prioridad de emergencia se suelen enviar a todos los usuarios que estén conectados al sistema, de manera que se den cuenta de que algo va mal:

.=emerg

Mostramos los mensajes urgentes a todos los usuarios conectados, mediante wall

*

1.7.2 Archivos de log

syslog

Este archivo se encuentra en /var/adm o /var/log y en él se guardan mensajes relativos a la seguridad del equipo, como los accesos o los intentos de acceso a ciertos servicios, esta información varia dependiendo de la configuración del archivo /etc/syslog.conf

```
Server1 :/# cat /var/log/syslog
Mar 5 04:15:23 server1 in.telnetd[11632]: connect from localhost
Mar 5 06:16:52 server1 rpcbind: connect from 127.0.0.1 to getport(R )
Mar 5 06:16:53 server1 last message repeated 3 times
Mar 5 06:35:08 server1 rpcbind: connect from 127.0.0.1 to getport(R )
Mar 5 18:26:56 server1 rpcbind: connect from 127.0.0.1 to getport(R )
Mar 5 18:28:47 server1 last message repeated 1 time
Mar 5 18:32:43 server1 rpcbind: connect from 127.0.0.1 to getport(R )
Mar 6 02:30:26 server1 rpcbind: connect from 127.0.0.1 to getport(R )
Mar 6 03:31:37 server1 rpcbind: connect from 127.0.0.1 to getport(R )
Mar 6 11:07:04 server1 in.telnetd[14847]: connect from server2
Mar 6 11:40:43 server1 in.telnetd[14964]: connect from localhost
```

Messages

Aquí se almacenan datos informativos de ciertos programas, mensajes de baja y media prioridad que informan de sucesos importantes, como el arranque del equipo, este archivo se encuentra en el directorio /var/adm

```
server1:/# head  -20  /var/adm/messages
Jan 24 18:09:54 server1 unix: SunOS Release 5.7 Version Generic
[UNIX(R) System V Release 4.0]
Jan 24 18:09:54 server1 unix: Copyright (c) 1983-1998, Sun Microsystems, Inc.
Jan 24 18:09:54 server1 unix: mem = 65152K (0x3fa0000)
Jan 24 18:09:54 server1 unix: avail mem = 51167232
Jan 24 18:09:54 server1 unix: root nexus = i86pc
Jan 24 18:09:54 server1 unix: IDE device at targ 0, lun 0 lastlun 0x0
Jan 24 18:09:54 server1 unix: piomode 0x280, dmamode 0x0, advpiomode
Jan 24 18:09:54 server1 unix: minpio 227, minpioflow 120
Jan 24 18:09:54 server1 unix: valid 0x6, dwdma 0x107, majver 0x0
Jan 24 18:09:54 server1 unix: PCI-device: ata@0, ata0
```

wtmp

Este archivo es binario y almacena información relativa a cada conexión y desconexión del sistema, su contenido se puede ver a través del comando last

```
server1:/# last -10
toni pts/11 localhost Mon Mar 6 11:07 - 11:07 (00:00)
toni pts/11 server2 Sun Mar 5 04:22 - 04:25 (00:03)
ftp ftp andercheran.aiin Sun Mar 5 02:30 still logged in
ftp ftp andercheran.aiin Sun Mar 5 00:28 - 02:30 (02:01)
ftp ftp server1 Thu Mar 2 03:02 - 00:28 (2+21:25)
ftp ftp server1 Thu Mar 2 03:01 - 03:02 (00:00)
ftp ftp localhost Thu Mar 2 02:35 - 03:01 (00:26)
root console Thu Mar 2 00:13 still logged in
reboot system boot Thu Mar 2 00:12
root console Wed Mar 1 06:18 - down (17:54)
```

Los registros guardados en este archivo y en utmp tienen la siguiente información: el nombre del usuario, el servicio que utiliza, desde donde se conecta y hora de acceso. Algunos otros Unix (como solaris o irix) ocupan un archivo wtmp extendido denominado wtmpx, con otros campos adicionales.

utmp

También es un archivo binario con información de cada usuario que está conectado en un momento dado, el programa /bin/login genera un registro en este archivo cuando un usuario se conecta, mientras que init lo elimina cuando se desconecta. Para ver el contenido de este archivo se utilizan los comandos last -f nombre_del_archivo, w y who.

```
server1:/# who
root console Mar 2 00:13
root pts/2 Mar 3 00:47 (unix)
root pts/3 Mar 2 00:18 (unix)
root pts/5 Mar 2 00:56 (unix)
root pts/6 Mar 2 02:23 (unix:0.0)
root pts/8 Mar 3 00:02 (unix:0.0)
root pts/7 Mar 2 23:43 (unix:0.0)
root pts/9 Mar 3 00:51 (unix)
root pts/10 Mar 6 00:23 (unix)
```

lastlog .

Es un archivo binario guardado en /var/adm y contiene un registro para cada usuario con la fecha y hora de su última conexión. Se pueden ver estos datos mediante la orden finger :

```
server1:/# finger toni
Login name: toni In real life: Toni at SERVER1
Directory: /export/home/toni Shell: /bin/sh
Last login Mon Mar 6 11:07 on pts/11 from localhost
No unread mail
No Plan.
```

faillog

Similar al anterior pero este guarda información sobre el último intento de acceso de cada usuario, una conexión es fallida si el usuario teclea incorrectamente su contraseña, esta información se muestra la siguiente vez que dicho usuario entre de manera correcta al servidor.

```
server1 login: toni
Password:
Linux 2.0.33.
1 failure since last login. Last was 14:39:41 on ttyp9.
Last login: Wed May 13 14:37:46 on ttyp9 from pleione.cc.upv.es.
```

loginlog

Si se crea (en Solaris) el archivo /var/adm/loginlog (que no existe), se registrarán dentro de éste los intentos fallidos de login, siempre y cuando se produzcan cinco o más de ellos de manera consecutiva.

```
server1:/# cat /var/adm/loginlog
toni:/dev/pts/6:Thu Jan 6 07:02:53 2000
toni:/dev/pts/6:Thu Jan 6 07:03:00 2000
toni:/dev/pts/6:Thu Jan 6 07:03:08 2000
toni:/dev/pts/6:Thu Jan 6 07:03:37 2000
toni:/dev/pts/6:Thu Jan 6 07:03:44 2000
```

btmp

En linux y HP-UX el archivo btmp se utiliza para registrar las conexiones fallidas al sistema, con un formato similar al que wtmp utiliza para las conexiones exitosas.

```
server1:~# last -f /var/adm/btmp |head -7
pnvarro ttyq1 term104.aiind.up Wed Feb 9 16:27 - 15:38 (23:11)
jomonra ttyq2 deportes.etsii.u Fri Feb 4 14:27 - 09:37 (9+19:09)
PNAVARRO ttyq4 term69.aiind.upv Wed Feb 2 12:56 - 13:09 (20+00:12)
panavarr ttyq2 term180.aiind.up Fri Jan 28 12:45 - 14:27 (7+01:42)
vbarbera ttyp0 daind03.etsii.up Thu Jan 27 20:17 still logged in
pangel ttyq1 agarcia2.ter.upv Thu Jan 27 18:51 - 16:27 (12+21:36)
abarra ttyp0 dtra-51.ter.upv. Thu Jan 27 18:42 - 20:17 (01:34)
```

sulog

Aquí se registran las ejecuciones de la orden su, indicando fecha, hora, usuario que ejecuta el su y el usuario cuya identidad adopta, terminal asociada y éxito (+) o fracaso (-) de la operación.

```
server1:/# head -4 /var/adm/sulog
SU 12/27 07:41 + console root-toni
SU 12/28 23:42 - vt01 toni-root
SU 12/28 23:43 + vt01 toni-root
SU 12/29 01:09 + vt04 toni-root
```

debug

En este archivo se registra información de depuración (debug) de los programas que se ejecutan en la máquina, esta puede ser enviada por las propias aplicaciones o por el núcleo del sistema operativo.

```
server1:~# tail -8 /var/adm/debug
Dec 17 18:51:50 server1 kernel: ISO9660 Extensions: RRIP_1991A
Dec 18 08:15:32 server1 sshd[3951]: debug: sshd version 1.2.21
[i486-unknown-linux]
Dec 18 08:15:32 server1 sshd[3951]: debug: Initializing random number
generator; seed file /etc/ssh_random_seed
Dec 18 08:15:34 server1 sshd[3951]: debug: Client protocol version 1.5; client
software version 1.2.21
Dec 18 08:15:34 server1 sshd[3951]: debug: Calling cleanup 0x800cf90(0x0)
Dec 18 16:33:59 server1 kernel: VFS: Disk change detected on device 02:00
Dec 18 23:41:12 server1 identd[2268]: Successful lookup: 1593 , 22 : toni.users
```

LOGS Remotos

El demonio syslog permite guardar registros en equipos remotos para registrar actividades sospechosas en un equipo seguro, esto se consigue definiendo un LOGHOST en lugar de un archivo normal en el /etc/syslogd.conf, por ejemplo, si queremos registrar la información de prioridad info y notice en un equipo remoto llamado server2, se indica así:

```
*.=info; *.=notice                @server2
```

En el equipo donde deseemos almacenar las bitácoras, se debe tener definido el puerto syslog en /etc/services y ejecutar syslogd con el parámetro '-r ' para que acepte conexiones a través de la red:

```
server1:~# grep syslog /etc/services
syslog 514/udp
server1:~# ps -fea|grep syslogd
root 41 0.0 0.4 852 304 ? S Mar21 0:01 /usr/sbin/syslogd
server1:~# kill -HUP 41
server1:~# syslogd -r
```

Todos los mensajes generados en el equipo origen se enviarán al equipo destino. Hay que tener cuidado, por que el tráfico en la red viaja en claro, y algún atacante con un sniffer podría interceptar este texto y tener acceso a información confidencial. Para evitar este problema existen 2 soluciones: se pueden registrar logs en un equipo directamente conectado al nuestro, sin emitir tráfico al resto de la red o bien utilizando comunicaciones cifradas como secure shell.

En el primer caso solo se requiere un equipo con 2 tarjetas de red, una por donde enviará el tráfico hacia la red local y la otra para conectarse con la máquina donde se almacenarán las bitácoras y solo podrá ser almacenada desde nuestro equipo y no tendrá usuarios, ni ofrecerá servicios.

El segundo caso se verá en la parte de secure shell y no es necesario que la máquina este aislada del resto de la red, ya que la transferencia de información se va a realizar de forma cifrada, para enviar un log cifrado a una máquina remota se puede utilizar SSH unido a las facilidades que ofrece syslogd, lo único que se necesita es tener un servidor sshd en la máquina destino y el cliente ssh en el origen.

Registros físicos

Es conveniente recurrir a registros físicos, ya que son mucho más difíciles de alterar que los lógicos. Unix permite almacenar mensajes en archivos especiales, como terminales o impresoras, estas últimas son las más habituales por la seguridad que brindan, ya que un intruso con privilegios puede modificar un archivo de bitácora pero no podrá eliminar información extraída por una impresora, al menos de que tenga acceso físico a la misma. El demonio syslog permite especificar estos archivos de dispositivo como destinatarios de los registros, solo se añade una entrada al archivo /etc/syslog.conf indicando el dispositivo y la clase de evento:

```
*.warn                            /dev/lp1
```

1.8 COMANDOS IMPORTANTES DE ADMINISTRACIÓN

A continuación se mencionan algunos comandos de administración que son de gran utilidad a cualquier administrador, principalmente para verificar fallas en los dispositivos de red o el origen de los problemas de comunicación que se presenten entre el servidor y el exterior, ya que estos son los más comunes.

1.8.1 ifconfig

Este comando se usa para configurar la(s) tarjeta(s) de red del servidor Unix, se indican parámetros como : dirección IP del servidor, máscara de red, dirección de broadcast, status de la tarjeta (up o down), etc.

Por ejemplo:

```
# ifconfig hme0
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500 inet 99.150.4.216 netmask
fffffc00 broadcast 99.150.7.255
```

Este comando nos sirve para poder detectar un mal funcionamiento de la tarjeta de red, como:

Dirección IP incorrecta : Es elemental poner la dirección IP elegida, ya que podríamos adquirir la de otro equipo y exponer el servidor a que no reciba los paquetes de manera correcta. Si alguien cambia la dirección del servidor, podría usarlo para realizar un ataque IP spoofing.

Dirección MAC incorrecta : Si la llegara a cambiar se utilizaría en forma similar al anterior, sólo que seria un ataque más elaborado, ya que se cambia la identidad no sólo de la IP, sino de la dirección MAC de la tarjeta, con esto sería más difícil detectar un ataque IP spoofing.

Tarjeta en modo promiscuo : Esto es, si instalan un sniffer en el servidor y ponen la tarjeta en modo promiscuo, se verán todos los paquetes que pasan por la red, ésta es la manera más sencilla de obtener nombres de usuarios y contraseñas.

1.8.2 route

Este comando se utiliza para configurar las tablas de ruteo del servidor, por lo menos existen 3 rutas, la de loopback (ocupa el dispositivo interno), la de red local (utiliza la tarjeta de red para comunicarse con equipos dentro del mismo segmento de red) y una default (que utiliza la tarjeta para enviar paquetes a un ruteador o un gateway que no son de la misma red). Si no se especifican parámetros , se muestra la configuración actual de las tablas de ruteo :

```
# route
Routing Table:
 Destination                    Gateway           Flags   Ref    Use     Interface
-----------------          ---------------        ----    ----   ------  --------
serv1                      99.150.4.1             UGH     0      0
10.150.37.180              99.150.4.4             UGHD    0      1
99.150.4.0                 serv1                  U       3      2483      hme0
224.0.0.0                  serv1                  U       3      0         hme0
default                    99.150.4.1             UG      0      13
localhost                  localhost              UH      0      199569   lo0
```

En algunos Unix esto se logra con el comando netstat –r.

Por medio de la tabla anterior se puede identificar alguna ruta incorrecta a pesar de que todas las demás estén funcionando correctamente, ya que no hay pérdidas de paquetes o retardos excesivos, sino solamente los paquetes podrían pasar por otro ruteador.

1.8.3 netstat

Este comando nos sirve para visualizar estructuras de datos del sistema de red, tablas de ruteo, estado de las conexiones al y desde el servidor, pasando por las tablas de ARP.

Respecto a la parte de seguridad, este comando nos muestra tablas de ruteo, puertos abiertos que escuchan peticiones de red y conexiones al servidor o desde él que se salgan de la normalidad. Un ejemplo muy útil del comando netstat :

```
# netstat -P tcp -f inet -a

TCP
  Local Address      Remote Address    Swind Send-Q Rwind Recv-Q State
-------------------- -------------------- ----- ------ ----- ------ -------
     *.*             *.*          0    0    0    0 IDLE
     *.sunrpc        *.*          0    0    0    0 LISTEN
     *.*             *.*          0    0    0    0 IDLE
     *.ssh           *.*          0    0    0    0 LISTEN
     *.shell         *.*          0    0    0    0 LISTEN
     *.exec          *.*          0    0    0    0 LISTEN
     *.32771         *.*          0    0    0    0 LISTEN
     *.32772         *.*          0    0    0    0 LISTEN
     *.fs            *.*          0    0    0    0 LISTEN
     *.32773         *.*          0    0    0    0 LISTEN
     *.dtspc         *.*          0    0    0    0 LISTEN
     *.32774         *.*          0    0    0    0 LISTEN
     *.7161          *.*          0    0    0    0 LISTEN
     *.*             *.*          0    0    0    0 IDLE
localhost.32775   localhost.7161    32768    0 32768    0 ESTABLISHED
localhost.7161    localhost.32775   32768    0 32768    0 ESTABLISHED
     *.9991          *.*          0    0    0    0 LISTEN
     *.32966         *.*          0    0    0    0 LISTEN
     *.*             *.*          0    0    0    0 IDLE
serv1.tnslsnr        *.*          0    0    0    0 LISTEN
     *.*             *.*          0    0    0    0 IDLE
     *.33248         *.*          0    0    0    0 LISTEN
     *.33249         *.*          0    0    0    0 LISTEN
     *.33250         *.*          0    0    0    0 LISTEN
     *.33251         *.*          0    0    0    0 LISTEN
serv1.33252       serv1.tnslsnr     32768    0 32768    0 ESTABLISHED
serv1.tnslsnr     serv1.33252       32768    0 32768    0 ESTABLISHED
serv1.33253       serv1.tnslsnr     32768    0 32768    0 ESTABLISHED
serv1.tnslsnr     serv1.33253       32768    0 32768    0 ESTABLISHED
     *.6000          *.*          0    0    0    0 LISTEN
serv1.ssh         12.153.5.180.3128 16456    0  8760    0 ESTABLISHED
     *.*             *.*          0    0    0    0 IDLE
```

Vemos los puertos abiertos (escuchando peticiones – LISTEN). Cualquiera puede conectarse a estos servicios o se puede evitar a través de TCP-Wrappers, como se detallará mas adelante. También se ven las conexiones establecidas entre el servidor y otros equipos (ESTABLISHED) o del servidor hacia el mismo, por lo tanto no hay problema como en el primer caso, si encontramos algo anormal como una conexión a través de telnet y desde un equipo desconocido, entonces se procede a monitorear esta conexión.

Otros datos que son de gran utilidad, son obtenidos cuando tenemos aplicaciones que ocupan puertos específicos o un rango de puertos. Esta salida nos muestra si realmente se están ocupando o si se establece la conexión.

1.8.4 ping

Este comando es usado comúnmente para verificar la comunicación con otro equipo o ver el estado del propio servidor, lo que hace es enviar paquetes ICMP (echo_request), que ocasionan que el sistema remoto también responda con paquetes ICMP, pero de tipo echo_response, si estos son recibidos quiere decir que el otro equipo esta encendido:

```
serv1:~# ping serv2
serv2 is alive
```

Algunos otros Unix pueden producir una salida mas detallada como la que se muestra:

```
serv2:~# ping -c 1 serv1
PING serv1 (195.195.5.3): 56 data bytes
64 bytes from 195.195.5.3: icmp_seq=0 ttl=255 time=0.2 ms
--- serv2 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
```

Este comando se puede utilizar como un arma para atacar a un servidor, un ejemplo es el ping flood, que consiste en saturar una red lenta con muchos paquetes ICMP, esto ocasiona una degradación del performance del servidor atacado. Para evitar problemas con paquetes ICMP es conveniente tener un firewall que cuide nuestro servidor o por lo menos registrar los paquetes ICMP que llegan al servidor (con la herramienta icmpinfo), aunque se debe tener cuidado con las bitácoras, ya que pueden llegar a saturar un filesystem crítico.

1.8.5 traceroute

Este comando se usa para ver la ruta que siguen los paquetes que salen del servidor hacia otro equipo; utiliza el campo TTL (time to live) del protocolo IP, inicializándolo con valores bajos y va aumentando conforme va recibiendo paquetes ICMP de tipo TIME_EXCEDED. La filosofía es sencilla, cada vez que un paquete pasa por un ruteador, éste se encarga de decrementar el campo TTL en una unidad, si se alcanza un valor igual a 0, se devuelve un paquete TIME_EXCEEDED y se descarta el paquete. Por lo tanto, traceroute inicializa este campo a 1, entonces el primer ruteador encontrado devuelve el mensaje de error, al recibirlo se inicializa a 2 y ahora el segundo ruteador descarta el paquete y envía otro mensaje de error, y así sucesivamente, de esta manera se construye la ruta hasta el equipo remoto deseado.

```
serv1:~# traceroute www.altavista.com
traceroute to altavista.com (204.152.190.70), 30 hops max, 40 byte packets
1 annex4.net.upv.es (158.42.240.191) 156.251 ms 144.468 ms 139.855 ms
2 zaurac-r.net.upv.es (158.42.240.250) 159.784 ms 149.734 ms 149.809 ms
3 atlas.cc.upv.es (158.42.1.10) 149.881 ms 149.717 ms 139.853 ms
4 A1-0-3.EB-Valencia1.red.rediris.es (130.206.211.185) 149.863 ms
150.088 ms 149.523 ms
5 A0-1-2.EB-Madrid00.red.rediris.es (130.206.224.5) 189.749 ms
159.698 ms 180.138 ms
6 A6-0-0-1.EB-Madrid0.red.rediris.es (130.206.224.74) 179.518 ms
7 194.69.226.13 (194.69.226.13) 259.752 ms 249.664 ms 259.83 ms
8 * * 195.219.101.1 (195.219.101.1) 290.772 ms
9 195.219.96.34 (195.219.96.34) 1680.33 ms 1660.36 ms 1669.83 ms
10 * 195.66.225.76 (195.66.225.76) 1660.68 ms 1650.33 ms
11 core1-linx-oc3-1.lhr.above.net (216.200.254.81) 2009.88 ms 1970.32 ms *
12 iad-lhr-stm4.iad.above.net (216.200.254.77) 2050.68 ms * *
13 sjc-iad-oc12-2.sjc.above.net (216.200.0.22) 2440.89 ms 2170.29 ms
14 pao-sjc-oc12-2.pao.above.net (207.126.96.65) 2441.19 ms 2140.32 ms *
15 mibh-above-oc3.pao.mibh.net (216.200.0.10) 2200.57 ms * *
16 * * www.altavista.com (204.152.190.70) 1810.56 ms
```

El comando traceroute se utiliza para realizar pruebas, medidas y administración de una red, el inconveniente es que introduce carga y puede afectarle al performance del servidor, inclusive puede llegar a negar servicios por el elevado tiempo de respuesta de las demás aplicaciones de red. Otro problema es que tiene permisos de setuid, por lo tanto se recomienda quitárselos.

1.9 UTILIZACIÓN DE RESPALDOS.

Antes de pasar al capitulo 2, en donde se detallará el funcionamiento de las herramientas que se han mencionado, es de suma importancia considerar como parte esencial de la administración de un servidor Uníx la realización continua de respaldos de información critica de nuestro servidor. Todo esto debido a que en el peor de los casos, si alguien logra entrar a un servidor con el objetivo de borrar toda nuestra información, con el respaldo realizado , se podrá recuperar la mayoria de la información.

Los respaldos son una excelente medida de prevención y recuperación de información crítica, y en la mayoría de las ocasiones el único mecanismo que tiene un administrador para restaurar un servidor. Se deben de realizar muy buenas estrategias (políticas) para realizar, almacenar y restaurar los respaldos, ya que se debe de recuperar la mayor cantidad de información posible, en el menor tiempo posible; otro punto importante es el horario que se ejecutarán los respaldos, ya que no se debe afectar el performance del equipo.

Un gran problema que se presenta cuando se realiza el respaldo es que no se verifica que éste haya sido exitoso, y cuando se intenta recuperar la información, se presenta un gran problema, por lo tanto se recomienda hacer pruebas de restauración. Otro problema que es muy común, son las etiquetas que se le ponen a los respaldos, estas deben ser breves y claras, para en caso de desastre se proceda a actuar de manera inmediata, sin necesidad de buscar cinta por cinta.

Las cintas que se generen de los respaldos deberán ser guardadas en un lugar muy seguro y no en el mismo lugar en donde se encuentran los servidores, ya que en caso de desastre por incendio o derrumbe, se perderían físicamente los servidores y las cintas.

Lo más importante a respaldar son los directorios en donde se realicen cambios cotidianos, por ejemplo, los directorios de los usuarios, los directorios en donde se localicen los sistemas, las bases de datos, etc. No es conveniente respaldar directorios como /usr/bin, /usr/sbin, /bin, /usr/lib o similares ya que el contenido de estos son archivos del Sistema operativo y estos los encontramos en el CD de instalación o en un respaldo que se haga de todo el equipo, éste se recomienda hacer solo una vez o cada que se haga un cambio relevante en la configuración del servidor.

1.9.1 Comandos para realizar respaldos

Muchos sistemas operativos traen herramientas de respaldo, aunque con esto se estaría limitando al momento de la restauración, ya que se necesitaría de esta herramienta obligatoriamente, por eso, es conveniente realizar los respaldos con los siguientes comandos, los cuales se ocupan en cualquier tipo de Unix:

dump/restore

El comando dump respalda filesystems completos y se recuperan con el comando restore. La sintaxis del comando es :

```
% dump opciones argumentos filesystem_a_respaldar
```

donde :

opción	Acción	Argumento
0 - 9	Nivel de la copia de seguridad	No
u	Actualiza /etc/dumpdates al finalizar el respaldo	No
f	Indica que se usará un cinta diferente a la de default	Si
b	Tamaño del bloque	Si
c	Indica que la cinta destino es un cartucho	No
W	Ignora todas las opciones, excepto el nivel del backup	No

En la tabla se ve que la opción "u" actualiza el archivo /etc/dumpdates, después de haber realizado un respaldo con éxito, dicho archivo deberá existir antes de ocupar dump por primera vez, en caso contrario no se almacenará información sobre los respaldos de cada filesystem. Información como, las copias de cada filesystem, su nivel y la fecha de realización, similar a:

```
/dev/dsk/c0d0s6 0 Thu Jun 22 05:34:20 CEST 2000
/dev/dsk/c0d0s7 2 Wed Jun 21 02:53:03 CEST 2000
```

Un ejemplo de la utilización de este comando es :

```
# dump [1-9]ucf cinta fs
```

Otro comando que sirve para respaldar de manera completa una partición lógica es el ufsdump, el cual manda información del respaldo en la pantalla:

```
# ufsdump 0cuf /dev/rmt /dev/dsk/c0d0s7

DUMP: Date of this level 0 dump: Thu Jun 22 10:03:28 2000
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/dsk/c0d0s7 (/export/home) to /dev/rmt
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 24523 blocks (118796KB)
DUMP: Writing 63 Kilobyte records
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: level 0 dump on Thu Jun 22 10:05:31 CEST 2000
DUMP: 24550 blocks (118927KB) on 1 volume
DUMP: DUMP IS DONE
```

Existe la facilidad de ejecutar respaldos remotos de la siguiente manera:

```
# ufsdump 0cuf user@hostname:/dev/rmt/0m /dev/dsk/c0d0s7
```

Si se utiliza el comando rdump, se debe definir un nombre de equipo denominado "dumphost" en el archivo /etc/hosts. Para cualquier comando que se use de manera remota (dump, ufsdump o rdump), el host remoto deberá tener como máquina de confianza en el /etc/hosts.equiv o .rhosts el equipo desde el cual se ejecuta el comando, con los problemas de seguridad que estos implican.

Para restaurar los respaldos realizados con dump se ocupa el comando restore el cual puede extraer archivos individuales, directorios o filesystems completos, la sintaxis es la siguiente :

```
# restore opciones argumentos archivos
```

donde opciones y argumentos es la misma lista que para el dump y archivos es la lista de directorios que se va a restaurar.

Opción	Acción	Argumento
r	Restaura la cinta completa	No
f	Indica el dispositivo o archivo donde esta el respaldo	Si
i	Modo interactivo	No
x	Extrae los archivos y directorios desde el directorio actual	No
t	Imprime los nombres de los archivos de la cinta	No

Para restaurar algunos archivos de un respaldo, guardado en el directorio backup, primero se puede consultar el contenido de la cinta :

```
serv1:~# restore -t -f backup > contenido
Level 0 dump of /home on serv1:/dev/hda3
Label: none
```

```
serv1:~# cat contenido

Dump date: Fri Jun 23 06:01:26 2000
Dumped from: the epoch
2 .
11 ./lost+found
30761 ./lost+found/#30761
30762 ./lost+found/#30762
30763 ./lost+found/#30763
30764 ./lost+found/#30764
30765 ./lost+found/#30765
30766 ./lost+found/#30766
30767 ./lost+found/#30767
4097 ./ftp
8193 ./ftp/bin
8194 ./ftp/bin/compress
8195 ./ftp/bin/cpio
8196 ./ftp/bin/gzip
8197 ./ftp/bin/ls
8198 ./ftp/bin/sh
8199 ./ftp/bin/tar
8200 ./ftp/bin/zcat
12289 ./ftp/etc
12290 ./ftp/etc/group
Broken pipe
```

Aquí ya se conoce el contenido del respaldo y los archivos a restaurar, se puede extraer un archivo en particular :

```
serv1:~# restore -x -f backup ./ftp/bin/tar
You have not read any tapes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for '.'? [yn] n
serv1:~# ls -l ftp/bin/tar
---x--x--x 1 root root 110668 Mar 21 1999 ftp/bin/tar
```

La restauración se hace a partir del directorio actual. Para extraer archivos a su ubicación original se debe de hacer desde el directorio adecuado. Otra opción del restore es la –i, para trabajar en modo interactivo, esta ofrece un prompt desde el cual se puede listar el contenido de la cinta, cambiarse de directorio o extraer archivos :

```
serv1:~# restore -i -f backup
restore > help
Available commands are:
ls [arg] - list directory
cd arg - change directory
pwd - print current directory
add [arg] - add `arg' to list of files to be extracted
delete [arg] - delete `arg' from list of files to be extracted
extract - extract requested files
setmodes - set modes of requested directories
quit - immediately exit program
what - list dump header information
verbose - toggle verbose flag (useful with ``ls'')
help or `?' - print this list
If no `arg' is supplied, the current directory is used
restore > ls
ftp/ httpd/ httpsd/ lost+found/ samba/ toni/
restore > add httpd
restore > extract
You have not read any tapes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for '.'? [yn] n
restore > quit
```

En el ejemplo anterior se consulta el contenido del respaldo, se selecciona el directorio httpd para extraerlo.

tar

Es la herramienta más sencilla para realizar respaldos, porque permite respaldar archivos individuales o directorios completos en un solo archivo, agrupa toda una estructura. Su desventaja es que si falla una parte de la cinta, se pierde todo el archivo de respaldo.

Opciones de tar:

Opción	Acción
c	Crea un archivo tar
x	Extrae archivos de un archivo tar
t	Verifica los archivos almacenados en un archivo tar
r	Inserta archivos al final de un archivo tar
v	Modo verbose
f	Especifica el nombre del archivo tar
Z	Comprime o descomprime mediante compress/uncompress
z	Comprime o descomprime mediante gzip
p	Conserva los permisos de los archivos

Por ejemplo para hacer un respaldo de /export/home/ en la unidad de cinta:

```
# tar cvf /dev/rmt/0m  /export/home
```

Otro ejemplo para respaldar muchos archivos o directorios, es indicándole que es lo que se quiere respaldar y dejándolo en un archivo tar en lugar de enviarlo a la cinta:

```
# tar cvf /tmp/resplado.tar  /etc/passwd /etc/host*
```

Para verificar el contenido de un respaldo:

```
# tar tvf /tmp/respaldo.tar
```

Para recuperar archivos de un archivo tar, se le puede indicar que archivo se desea recuperar :

```
# tar xvf /tmp/respaldo.tar etc/passwd
```

ó en caso contrario restaura todo :

```
# tar xvf /tmp/respaldo.tar
```

La restauración se realiza desde el directorio de trabajo actual, creándose un subdirectorio **/etc** debajo del actual, con los archivos correspondientes. Si los archivos se desean poner en el lugar de origen, se deben restaurar en el directorio adecuado.

cpio

cpio es otra utilidad que permite respaldar archivos a o desde un archivo cpio, que no es más que un archivo que almacena otros archivos e información sobre estos (permisos, nombres, propietario, etc). Este archivo cpio puede ser un disco, un archivo, una tubería o una cinta, mientras que los archivos a copiar pueden ser archivos normales o filesystems.

Opciones de cpio :

Opción	Acción
o	Realiza la exportación de archivos
i	Realiza la importación de archivos
m	Conserva fecha y hora de los archivos
t	Crea tabla de contenidos
A	Añade archivos a un archivo cpio existente
v	Modo verbose

La sintaxis es un poco más compleja, debido a las opciones de adentro y afuera, esto es, afuera es generar un archivo en la salida estándar, mientras que adentro es lo contrario, es decir, extrae archivos de la entrada estándar.

Por ejemplo para copiar los archivos de /export/home en el archivo /tmp/backup.cpio :

```
# find /export/home | cpio –o > /tmp/backup.cpio
```

cpio lee de la entrada estándar, esperando los nombres de los archivos a guardar, por lo tanto, se utiliza una tubería. Se debe redireccionar su salida al nombre del archivo cpio, por que si no solo se mostraría en la salida estándar (regularmente la terminal). Es recomendable usar find ya que este muestra la ruta completa de los archivos y/o directorios.

Para verificar el contenido de un respaldo, se utiliza :

```
# cpio –t < /tmp/backup.cpio
```

Para la restauración de un archivo, se le debe especificar el nombre de éste a través de una tubería (similar a la del respaldo) :

```
# echo " /export/home/fabian/temporal.txt" | cpio -i < /tmp/backup.cpio
```

Para restaurar todo :

#cpio –i

2. CAPÍTULO 2 : CARACTERÍSTICAS Y APLICACIONES DE LAS HERRAMIENTAS DE SEGURIDAD DE LIBRE DISTRIBUCIÓN.

2.1 INTRODUCCIÓN

En este capítulo se revisarán solo las herramientas de seguridad que son de libre distribución, por lo tanto, están disponibles para que sean usadas por cualquier administrador.

Existen muchas herramientas de libre distribución, pero en esta tesis solo se especificarán y evaluarán algunas, y en capítulos posteriores se verá el uso adecuado de ellas en conjunto o de manera individual.

Se detallarán varias herramientas para diferentes propósitos y usos variados, sus características, el uso adecuado que se le debe dar y las ventajas y desventajas de cada una.

2.2 TECNOLOGÍAS DE AUTENTICACIÓN DE ACCESO REMOTO

2.2.1 npasswd

npasswd es un reemplazo del comando passwd de Unix. Este programa obliga al usuario a usar passwords que sean difíciles de adivinar, esto limita a los usuarios de escoger passwords débiles. npasswd trabaja en muchas versiones de Unix, pero no tiene todas las características de cada programa passwd de todas las plataformas Unix. Se debe tener mucho cuidado el tomar la decisión de cambiar el comando passwd por la herramienta npasswd.

Muchas plataformas de Unix usan el /etc/shadow, lo que dificulta a un atacante obtener los passwords cifrados, aunque la mayoría de los sistemas aún tienen el archivo /etc/passwd disponible para todos los usuarios del sistema.

Teniendo el archivo /etc/shadow reduce la vulnerabilidad de adivinar los passwords, pero si se esta usando NIS o NIS +, se vuelve vulnerable el sistema. La combinación de passwords fuertes y la base de datos del shadow, proveé la mejor protección para el uso de passwords. La instalación y la configuración de npasswd no es compleja, se detallará en el siguiente capítulo.

2.3 CIFRADO DE DATOS

2.3.1 PGP (Pretty Good Privacy)

PGP es una herramienta de cifrado, en la que 2 llaves (pública y privada) son usadas para entablar comunicaciones seguras. La llave privada pertenece solo al propietario y la llave pública se puede distribuir libremente a otros usuarios de PGP. Estas llaves son guardadas en archivos de llaves (llaveros).

La mayor funcionalidad de PGP es enviar mensajes de correo electrónico cifrados y también firmar o cifrar archivos personales. Todo esto con la finalidad de evitar que se altere información enviada e interceptada a través de la red.

Un ejemplo de su uso es el siguiente : el emisor al enviar un archivo o un correo privado a otro usuario (receptor), usa una copia de la llave pública de éste para cifrar la información, la cual solo podrá ser descifrada usando la llave privada del receptor, y de manera inversa es exactamente lo mismo.

Un usuario puede usar su llave privada para firmar su correo electrónico o sus archivos para autenticarlos, entonces los destinatarios usan una copia de la llave pública de este usuario para asegurarse de que es realmente el que lo envió y que no haya sido alterado durante su trayecto.

Una vez que se haya instalado PGP, se puede empezar a intercambiar información de manera segura con otros usuarios de PGP. Obviamente es necesario tener las llaves públicas de ellos y distribuir la llave pública personal. La llave pública es un bloque de texto, por lo tanto el intercambio de llaves publicas es muy sencillo, se puede mandar a través de un archivo, un correo o publicarla en un servidor de llaves públicas. Al obtener una llave pública de alguna persona se puede poner en un archivo de llaves públicas.

2.4 COMUNICACIÓN

2.4.1 Secure shell (ssh)

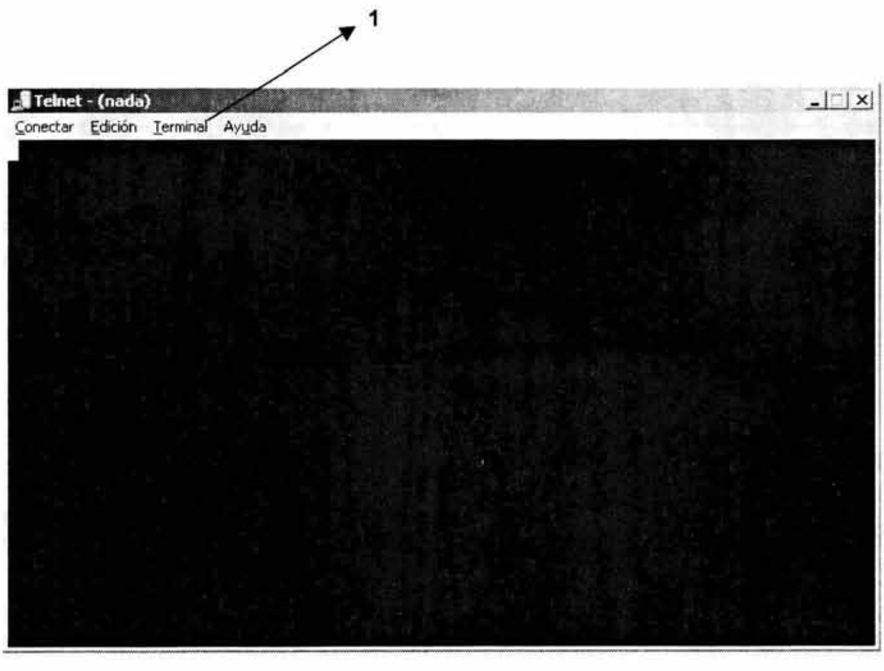
Es una herramienta que permite realizar una conexión entre equipos, ejecuta programas remotos y transfiere archivos, todo esto de manera segura y confiable. Secure shell viene a reemplazar los servicios más fáciles de atacar en un servidor : telnet, ftp, rlogin, rsh y rexec. Estos servicios son los primeros que monitorea un atacante para intentar acceder a un equipo, ya que con un simple sniffer se puede visualizar lo que está "viajando" [1] a través de la red, como estos son los más usuales, por su fácil y cómodo manejo. Otra facilidad de secure shell es que permite enviar datos seguros con un ambiente gráfico como X-windows.

El uso de secure shell es muy sencillo, ya que es muy similar a telnet. La primera vez que se establece la conexión es cuando se realiza el intercambio de llaves, la autenticación y el cifrado, el cual es usado en sesiones posteriores, pero sin que el usuario se percate.

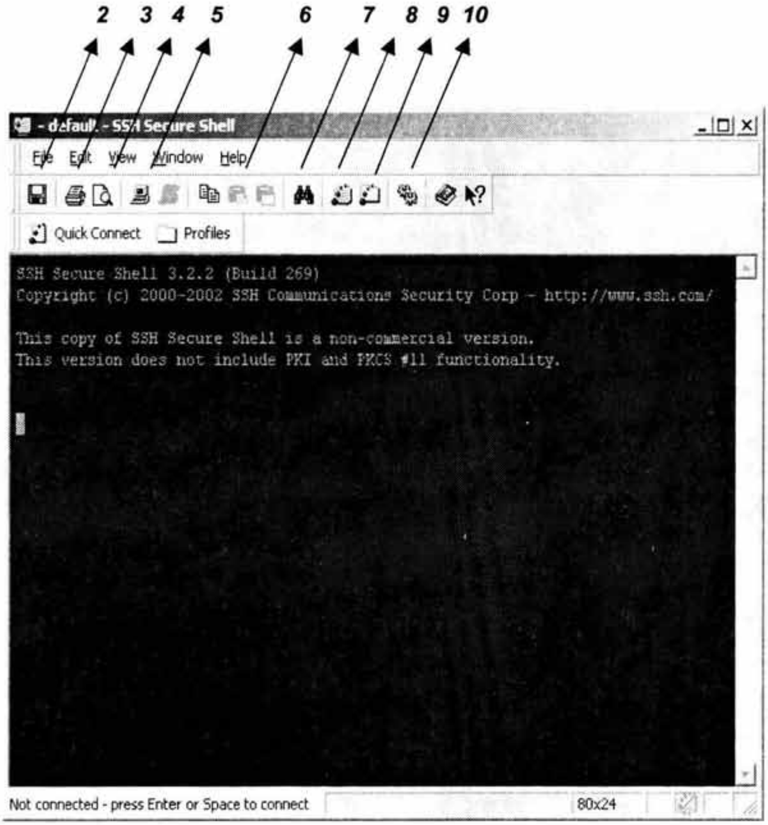
Se desarrollaron 2 protocolos sobre ssh:

SSH1: Esta se utiliza para propósitos no comerciales y generalmente la ocupan las Universidades.
SSH2: Esta es comercial y necesita de licencia para poder trabajar.

Comparación entre clientes para terminales (telnet y secure shell)



1 **Este término es usado cuando se intenta decir que la información que se está transfiriendo por la red, puede ser visualizada por algún software.**

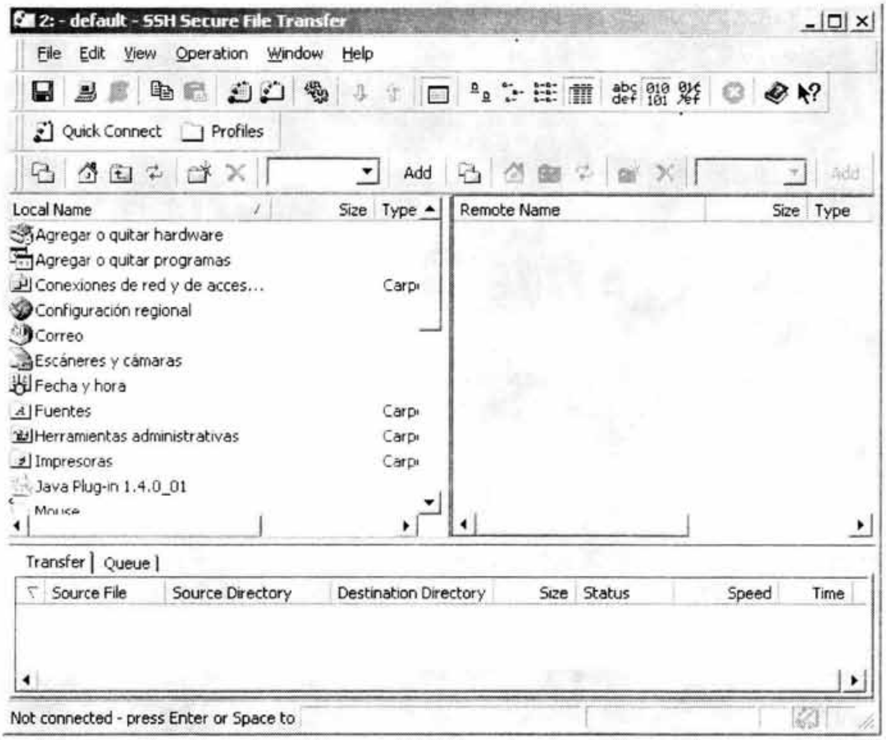


1. **Barra de menús (limitada) en modo texto**

Para secure shell, la barra de herramientas es gráfica.

2. **Guarda los cambios a las características de la terminal.**
3. **Imprime lo que se ha realizado en esta sesión.**
4. **Visualiza lo que se imprimirá.**
5. **Establece la conexión con el servidor deseado.**
6. **Facilita el copy-paste tan usado en la actualidad.**
7. **Hace una búsqueda en todo lo que se ha efectuado.**
8. **Permite abrir mas terminales sin teclear la contraseña en cada caso.**
9. **Abre una sesión gráfica del ftp seguro (se muestra en la siguiente gráfica)**
10. **Aquí se modifican las características de la terminal.**

FTP Gráfico



Esta es el ftp gráfico de secure shell, por lo tanto, también es seguro. A simple vista se nota que es muy similar al explorador de Windows, lo que hará mucho mas fácil su uso.

2.4.2 Openssh

Openssh es un proyecto, desarrollado por OpenBSD. El Openssh tiene las mismas funciones que secure shell, la diferencia es que este es una versión libre, la cual es compatible con los protocolos SSH1 y SSH2. Esta herramienta se está utilizando demasiado y está desplazando a secure shell, debido a que es de libre distribución, la única desventaja que tiene es que con pruebas que se le realizaron, demostró inestabilidad (sobretodo en la transferencia de archivos), entonces es recomendable actualizar sus versiones y estar enterado de las vulnerabilidades que se presenten con la versión que se tenga instalada [2].

2 Fuente: " http://www.openssh.com/"

2.5 MONITOREO DE RED

2.5.1 TCP-wrappers

La herramienta TCP-wrappers sirve para monitorear el tráfico que llega al servidor, principalmente para proteger los sistemas y detectar rarezas dentro de éste. El objetivo de TCP-Wrappers es proteger los programas o servicios dentro del sistema, logrando un mayor nivel de seguridad.

El TCP-Wrappers nos ayuda para simular o cambiar la manera en que el servidor responderá, pero sin que afecte su performance, ni su forma de trabajar, es decir, solo servirá como una protección a los servicios de red que se brindan, sin que se note un retardo y sin que se modifiquen las aplicaciones o la manera de conectarse a ellas.

Tiene muchas ventajas esta herramienta, por su fácil instalación, configuración y uso, además de trabajar de manera independiente, pero sin dejar de tener relación con los servicios o programas del servidor.

A continuación se describe la forma de trabajar de TCP-Wrappers [3] :

En un servidor siempre está corriendo un demonio, el cual está esperando cualquier petición de conexión de red que se le haga. En este caso el demonio inetd es el que recibe la petición y otorga el servicio apropiado, para volver a esperar otra petición y así sucesivamente.

La labor de TCP-Wrappers es recibir todas las peticiones que se le hagan al demonio del inetd y primero validarlas y registrarlas en bitácoras, si está permitida la acepta, si no la rechaza, con la facilidad de poder enviarle alguna respuesta al emisor.

Se compone de 5 programas:

tcpd Demonio de TCP-Wrappers.
tcpdmatch Define como el tcpd manejaría una petición de conexión
tcpdchk Verifica el control de acceso por medio de los archivos /etc/hosts.allow y /etc/hosts.deny.
safe-finger Similar al comando finger, para implementar el finger respuesta.
try-from Verifica si el sistema es capaz de reconocer que máquina la está contactando.

2.5.2 Sniffer (como herramienta analizadora)

En el ambiente de redes todo se transmite por medios compartidos y la información viaja en claro por estos medios, por lo tanto, cualquier equipo puede capturar lo que se transmite (sniffing). Como la mayoría de las redes trabajan con Ethernet se facilita su acceso, debido a que el paquete ethernet contiene la dirección de la máquina destino, se supone que con esto se limita a que solo ese equipo pueda recibir dicho paquete, sin embargo, si un equipo pone su tarjeta en modo promiscuo, significa que podrá ver todos los paquetes que pasen por la red.

El sniffer es un programa que puede monitorear y analizar el tráfico de la red, si se hace un buen uso de este programa, puede servir para detectar cuellos de botellas y posibles colisiones, pero también podrá ser utilizado para captar datos confidenciales y que pueden ser empleados por alguna persona malintencionada y dañar los datos de algún usuario, incluyendo a root.

3 Fuente: http://www.asc.unam.mx/

2.5.3 Portsentry

Una de las maneras de obtener información de un servidor o equipo es a través de los barridos de puertos, esto se hace mediante intentos de conexión a todos los puertos, descubriendo cuales están abiertos y por lo tanto disponibles. Esto se realiza mediante herramientas o programas sofisticados que se encargan del barrido de puertos y que son difíciles de detectar. Existe una herramienta que permite encontrar patrones utilizados al rastrear puertos y también puede ejecutar acciones correctivas.

Porsentry es una herramienta que se encarga de monitorear los puertos que se consideren que deben estar inactivos, cuando llega una conexión a uno de estos puertos, se marca en una bitácora, se bloquea la comunicación con la máquina que está realizando el scanneo y se ejecuta una acción (algún comando).

Su manera de trabajar es la siguiente : se le indica que escuche determinados puertos (TCP y UDP) que sean poco solicitados, los cuales son especificados en las opciones UDP_PORTS y TCP_ PORTS en el archivo de configuración. Portsentry abre el socket de los puertos especificados sin proporcionar algún servicio, registrando cualquier intento de conexión y ejecutando una acción o comando.

Otro modo es abriendo los sockets crudos, con esto se pueden detectar más ataques de conexión y barridos de puertos. También existe otra manera en donde no se abre ningún puerto, sino que el kernel notificará si llega alguna petición a algún puerto menor al especificado en opciones del archivo de configuración, la desventaja es que se escuchan puertos por el kernel y esto requiere una configuración especial, si se realiza mal se provocaría una negación de servicios. Con esta opción aparentemente los puertos no están escuchando, porque no están abiertos. Las opciones ADVANCED_EXCLUDE_TCP y ADVANCED_EXCLUDE_UDP del archivo de configuración permiten excluir puertos que sean muy solicitados.

Como se mencionó anteriormente corre en sockets TCP y UDP para detectar scanneos de puertos a un servidor. Portsentry es configurable para correr en múltiples sockets al mismo tiempo. Detecta barridos de puertos y responde con un bloqueo a la máquina que está haciendo el ataque. También tiene una parte donde aloja los equipos que se han conectado, para identificar falsas alarmas de algún intento de barrido de puertos.

Reporta todas las violaciones al equipo local o a un equipo remoto indicando el nombre del sistema, hora del ataque, IP del equipo atacante y el puerto TCP o UDP que haya sido scanneado. Cuando portsentry detecta un scanneo al sistema genera un bloqueo hacia el otro sistema, y éste no lo detecta, por lo tanto no se percatará que ya no podrá realizar ninguna otra acción.

2.6 MONITOREO DEL SISTEMA

2.6.1 Cops

La función de esta herramienta es detectar posibles huecos de seguridad como : vulnerabilidades en archivos binarios, archivos con SUID, programas que estén en los directorios rc's, las tablas de cron, seguridad en archivos /etc/group y /etc/passwd, cuentas sin password, passwords débiles y permisos comprometedores en archivos del sistema.

Cops tiene una serie de programas que de manera automática detecta vulnerabilidades en un sistema, documentación y scripts de instalación. Una vez que se instala la herramienta (como se verá en el siguiente capitulo), cops generará reportes dentro de un directorio, el cual tendrá el nombre del equipo y dentro de éste estarán los reportes con formato de año_mes_día.

Cops es una colección de herramientas seguras que se diseñaron especialmente para auxiliar a los administradores de sistemas Unix. La función de cops es la de advertir los problemas que se encuentren, pero no corrige ni explota cualquiera de los problemas encontrados. Una opción seria generar shells scripts que contengan la solución a los problemas encontrados, pero deberá ser realizado con mucho cuidado y probado, por que regularmente los problemas encontrados solo los puede resolver el usuario root y se podría modificar o incluso borrar información crítica y necesaria para el buen funcionamiento del servidor.

Otra ventaja de cops es que no tiene que ser ejecutado necesariamente por root, a menos que se requiera localizar los archivos con SUID en todo el sistema. Cops no podrá ser ejecutado de manera remota, solo localmente.

Cops brinda un método de verificación de errores comunes, esto es, no resuelve todos los problemas de seguridad, solo se le deberá considerar como una utilería mas de ayuda, para proteger al sistema de los usuarios o del propio dueño, tal vez por ignorar algunos problemas con la seguridad.

Cops no corrige errores directamente, por diferentes razones, la principal es que la seguridad es muy variada y depende de lo que se quiera proteger. Es probable que intencionalmente se dejen abiertos algunos servicios que puedan ser vistos como huecos u otorgar permisos a algunos archivos para que puedan ser accedidos sin problema alguno.

Entonces se considera una herramienta que hace más sólida la seguridad, y no se considera un arma que pueda ser usada por alguien que quiera atacar un sistema, aunque si se debe de tener cuidado de quien pueda ejecutar esta herramienta.

2.6.2 Tripwire

El objetivo de esta herramienta es verificar la integridad de los sistemas Unix, verificando cambios en la estructura de los archivos o directorios, usando firmas digitales y si existe algún cambio raro o inesperado se podrá determinar si alguien realizó esto con el fin de perjudicar al sistema u obtener algo de él.

Se le considera como un monitor de integridad, ya que usa rutinas de checksum, huellas y firmas para detectar cambios realizados al sistema, por ejemplo, verifica : cambios en permisos de archivos, ligas, cambio en el tamaño de archivos críticos, UID's y GID's.

Tripwire es un programa que se encarga de verificar la integridad de los filesystems. Para poder realizar esto, primero se deberá construir un archivo de configuración en el cual se especifica a que directorios y archivos se les verificarán sus atributos. Al correr tripwire se crea un base de datos con los directorios y archivos especificados.

Además de verificaciones (checksums) que se realizan a cada directorio y archivo, la base de datos de tripwire también contiene información que permite verificar :

- Permisos en archivos (principalmente de ejecución)
- Número de inodos en los file systems
- Número de ligas
- GID del grupo de usuarios que tengan privilegios sobre el sistema
- Tamaño de los archivos
- Fecha y hora de su último acceso, la última modificación y la hora y fecha de creación.

Para cualquier sistema se recomienda revisar la integridad de los archivos del sistema operativo y demás archivos que se consideren críticos y que no tengan que cambiar sobre condiciones normales de trabajo, pero sin dejar de revisar archivos ejecutables, demonios, scripts , librerías y archivos de configuración asociado con éstas.

Al momento de escoger que archivos se les verificarán sus atributos, es necesario considerar cuales son usados regularmente por el sistema, como podrían ser las bitácoras, que son regularmente actualizadas. Entonces, no conviene monitorear el incremento de su tamaño, pero tal vez si monitorear el tamaño de los archivos binarios que escriben en las bitácoras o permisos de las mismas bitácoras.

Es necesario hacer una afinación en la configuración de tripwire, para poder determinar un efectivo archivo de configuración que solo tenga lo necesario y quitar mensajes que se estén generando innecesariamente, esto solo se logrará con la revisión constante de los mensajes generados y así poder decidir que es lo que se monitoreará. Para que se pueda instalar tripwire en un sistema es necesario que tenga un compilador de C o gcc.

Ejemplos de algunos reportes que se pueden generar con tripwire, si no encuentra cambios durante la verificación mandará una salida similar a esta :

```
### Phase 1: Reading configuration file
### Phase 2: Generating file list
./tripwire: /etc/dfs/sharetab: No such file or directory
./tripwire: /etc/hosts.equiv:  No such file or directory
./tripwire: /etc/named.boot: No such file or directory
./tripwire: /etc/rmtab: No such file or directory
### Phase 3: Creating file information database
scanning: /etc scanning: /etc/TIMEZONE
scanning: /etc/aliases
scanning: /etc/autopush
scanning: /etc/clri
scanning: /etc/crash
scanning: /etc/cron
scanning: /etc/cron.d
scanning: /etc/defaultrouter
scanning: /etc/resolv.conf
scanning: /etc/shadow
scanning: /etc/one-time.sh
scanning: /etc/mkgroup
scanning: /etc/mkpasswd
scanning: /etc/mkshadow
### Phase 4: Searching for inconsistencies
### ### Total files scanned:    461
### Files added:          0
### Files deleted:        4
### Files changed:          454
### After applying rules:
### Changes discarded:        454
### Changes remaining:         0
```

En la última fase de la revisión entrega los resultados.

En el siguiente ejemplo se reportan modificaciones a archivos :

```
### Phase 1: Reading configuration file
### Phase 2: Generating file list
./tripwire: /etc/hosts.equiv: No such file or directory
./tripwire: /etc/rmtab: No such file or directory
### Phase 3: Creating file information database
### Phase 4: Searching for inconsistencies
### Total files scanned: 463
### Files added:       2
### Files deleted:     4
### Files changed:      456
### After applying rules:
### Changes discarded:  455
### Changes remaining:  5
added: -rw------- root  0 Apr 19 16:21:14 1997 /etc/.pwd.lock
added: -rw------- root  0 Apr 19 16:21:13 1997 /etc/.group.lock
changed: drwxr-xr-x root 3584 Apr 19 16:21:14 1997 /etc
changed: -r--r--r-- root 6982 Apr 19 16:20:58 1997 /etc/passwd
changed: -r-------- root 1571 Apr 19 16:20:59 1997 /etc/shadow
### Phase 5: Generating observed/expected pairs for changed files
### Attr Observed (what it is) Expected (what it should be)
  st_mtime: Sat Apr 19 16:21:14 1997  Sat Apr 19 14:00:09 1997
  st_ctime: Sat Apr 19 16:21:14 1997  Sat Apr 19 14:00:09 1997
/etc/passwd
  st_ino: 74339                74305
  st_size: 6982                6932
  st_mtime: Sat Apr 19 16:20:58 1997  Sat Apr 19 14:00:08 1997
  st_ctime: Sat Apr 19 16:20:58 1997  Sat Apr 19 14:00:08 1997
 md5 (sig1): 1k0No.UNC9mCJgIGMtPk4O   0D5zziXYpwvXxOzy.DrYCx
/etc/shadow
  st_ino: 74305 74338
```

Como se nota las primeras 2 líneas de la fase 4, reporta que algunos archivos fueron creados, las 3 líneas siguientes muestran los cambios que se le aplicaron a directorios y archivos existentes y en la fase 5 reporta modificaciones en el directorio /etc, en atributos del archivo /etc/passwd y el número de inodo del /etc/shadow.

2.6.3 Saint

Es una herramienta que recolecta información de equipos remotos para examinarle servicios de red como : finger, NFS, NIS, ftp, tftp, rexd, statd y otros servicios. La información recopilada incluye la presencia de varios servicios de red que se consideran huecos de seguridad, esto generado regularmente por una mala instalación, mala configuración de los servicios de red o por aplicar políticas incorrectas al sistema. El saint puede generar un reporte simple o a detalle basado en reglas que se le configuren, se puede examinar, manipular y analizar la salida con un navegador como explorer o netscape. Dado que el programa está orientado principalmente a analizar resultados de implicaciones de seguridad, mucha información obtenida puede ser usada para ver los servicios de red que están corriendo, tipo de hardware utilizado y software que se está usando en la red, similar a un tipo de inventario.

La herramienta se explota al usarla como exploradora de los servidores, basado en una colección de datos y reglas de configuración. Ésta se encarga de examinar los caminos confiables para una buena configuración así como sus dependencias, para que se puedan aplicar en otros equipos. Saint no solo permite analizar los equipos de un administrador, sino todos los equipos en la red y además como estos podrían perjudicarla, entonces ayuda a tener un mayor nivel de seguridad a nivel de red, con todos los equipos involucrados.

Es una herramienta muy útil para que los administradores aseguren sus sistemas, sin embargo como es una herramienta de libre distribución, cualquier persona puede instalarla y usarla, entonces podría brindar información de los huecos que existen en otros servidores, incluso obtener información de vulnerabilidades que ni siquiera el administrador del sistema conoce.

La manera en que trabaja es la siguiente. Primero determina que equipos están disponibles en la red, entonces ejecuta una serie de pruebas y entrega los resultados obtenidos. Cada host es examinado y se obtiene un registro con el nombre del equipo, la prueba ejecutada y todos los resultados encontrados, todo esto es guardado para realizar un análisis. En el browser se ven ligas de todos los datos obtenidos de manera comprensible y detallada, mostrando los problemas y probables soluciones si existen.

2.7 OTRAS HERRAMIENTAS

2.7.1 Sudo

Sudo (superuser do) permite a un usuario ejecutar comandos de superusuario, solo si está en el archivo /etc/sudoers. Sudo requiere que el usuario se identifique con su propia contraseña, en este momento se le asigna un timestamp para que el usuario pueda usar sudo durante 5 minutos sin que le pida contraseña de nuevo.
Permite a los administradores del sistema otorgar privilegios a ciertos usuarios para que puedan ejecutar comandos de administración (que solo root puede ejecutar), entre sus características más importantes están:

- Restringir que comandos puede correr un usuario **X**
- No es necesario ponerle al usuario **X** el UID = 0
- Al momento de que el usuario ocupa algún comando permitido, se le pedirá la contraseña (de sistema operativo) para mayor seguridad.
- Todo el control de los comandos permitidos se lleva a través de un archivo de configuración.

Desventajas sin la herramienta SUDO:

- Usuarios con UID=0 (no incluyendo la cuenta de root), por lo tanto, con privilegios de administrador.
- Por ejemplo :
  adm:x:0:
- UID=0 Con este hueco de seguridad se podrían saltar los métodos usuales de identificación para realizar cualquier tarea de administración en el servidor.
- Si la conexión es vía telnet, el password de la cuenta viaja en claro por la red

Ventajas con la herramienta SUDO:

- Proteger la integridad del servidor
- Solo existirán usuarios "normales", con los respectivos privilegios
- No se restringe el acceso a esta cuenta
- Se lleva un mejor control en la administración
- Las cuentas no tendrán UID = 0, lo que las hace menos vulnerables
- Se puede distribuir la administración, entre varios usuarios.

2.7.2 Nullshell

El nullshell es un programa que sustituye a cualquier shell y como su nombre lo indica es un shell inválido. Esta herramienta fue diseñada para poder desactivar una cuenta de sistema operativo Unix, sin borrar nada de ésta, sin cambiarle su contraseña, conservando sus archivos y su estructura.

Otros usos serían :

- Solo desactivar una cuenta, sino se desea saber quien la está utilizando.
- Si solo se desea registrar sus intentos de acceso.
- Desactivar la cuenta de manera temporal sin borrar su estructura.

La forma de usar esta herramienta es poniendo la ruta del programa (nullshell) en lugar del shell en el archivo /etc/passwd, entonces cuando un usuario se conecta al servidor, se le mostrará un mensaje, el contenido de este dependerá del administrador del sistema. En este momento se registra el intento de conexión en un archivo, en donde se indica cuando se conectó y desde que máquina intentó establecer la sesión.

Se pueden cambiar los siguientes parámetros antes de compilar la herramienta :

- La variable FOAD_LOG, aquí se pone el archivo donde se alojarán los registros de intentos de conexión, por default viene : /usr/adm/foad.log.
- También se puede cambiar el mensaje que se mostrará en la cuenta que se deshabilitó, el cual está en un printf.

La compilación del programa que está escrito en lenguaje C se ejecuta así:

```
# cc -o nullshell nullshell.c
```

Con esto se obtiene el archivo ejecutable, el cual se debe colocar en donde están los demás shells o en otro lugar, con dueño root y con el SUID activado. Con esto ya se puede poner en cualquier cuenta, modificándola en el archivo /etc/passwd, poniendo la ruta exacta de su ubicación y la cuenta quedará inmediatamente desactivada y monitoreada.

Algo muy importante para que funcione adecuadamente es poner el nullshell en el archivo /etc/shells.

3. CAPÍTULO 3 : INSTALACIÓN Y CONFIGURACIÓN DE LAS HERRAMIENTAS DE SEGURIDAD DE LIBRE DISTRIBUCIÓN.

3.1 INTRODUCCIÓN

En el capítulo anterior se dio una semblanza de la funcionalidad de las herramientas propuestas para usar en la tesis, en este capítulo se hará de manera práctica lo que se había visto teóricamente, ya que se indicará como instalar y configurar cada una de las herramientas mencionadas. Sólo se explicará la instalación y configuración básica de cada herramienta, ya que el objetivo no es detallar la configuración, sino, identificar que se puede obtener de cada una para el propósito de esta tesis. Será una gran ayuda para cualquier administrador este capítulo, solo les restará leer un poco más para realizar una configuración avanzada.

En algunas instalaciones habrá detalles o problemas al momento de instalarlas, pero aquí se señalarán los pasos para que estos problemas no se presenten a los administradores. Se utilizará el método de compilación de la herramienta, aunque en algunos sitios de internet se distribuyan ya compiladas o en algunos casos vengan empaquetadas según el sistema operativo que se esté usando. Se recomienda que se ocupe el método de compilación, ya que resulta más fácil resolver problemas presentados durante la instalación.

Las instalaciones se deberán de realizar de preferencia con el usuario root, salvo algunos casos, en donde lo especifique la herramienta a instalar

Todas las herramientas deben ser bajadas de un sitio seguro en internet (ftp), después descomprimirlas (gunzip) :

```
# gunzip nombre_de_la_herramienta.gz
```

y descompactarlas con el comando tar, la sintaxis está definida en el capítulo 1

3.2 TECNOLOGÍAS DE AUTENTICACIÓN DE ACCESO REMOTO

3.2.1 npasswd

La instalación de npasswd consiste de lo siguiente :

```
# gunzip npasswd-2.05.tar.gz
# tar xvf npasswd-2.05.tar
```

Correr el script Configure, el cual es un shell que determinará el ambiente del sistema para poder construir npasswd. Definirá que clase de sistema está corriendo, características y tipo de compilador.

Después de correr Configure se obtiene otro shell llamado config.sh, el cual contiene variables de ambiente que serán puestas en otros archivos. Cuando se instala por primera vez, es conveniente no modificar los shell's y ejecutarlos con su configuración que traen por default. Algo en donde siempre se debe tener mucho cuidado es en los permisos de los shell's, verificar que tenga los de ejecución ya que regularmente es el primer punto de falla que se presenta.

```
# cd ./npasswd-2.05
# ./Configure
```

Este script va realizando una serie de preguntas, tomando las respuestas por default es suficiente :

```
Use which C compiler? [cc] gcc
Do you expect to run these scripts and binaries
on multiple machines? [n] n
Where will private files be installed? (~name ok)/usr/lib/passwd]
/usr/lib/passwd
Directory /usr/lib/passwd  doesn't exist. Use that name anyway? [n] y
### Found passwd files "/etc/passwd"
Change passwd file list? [n] n
### Found shadow files "/etc/shadow"
Change shadow file list? [n] n
Replace system programs? [y] y
Activate the "paranoid" open [n] y
Password history file [/usr/lib/passwd/history]
/usr/lib/passwd/history
```

Después de haber armado la configuración, se procede a la compilación del software con el comando make :

```
# make
```

El siguiente paso es la instalación de npasswd :

```
# make install
```

Para inicializar la base de datos histórica de npasswd

```
# /usr/lib/passwd/history_admin load < /dev/null
```

Se generan la siguiente estructura :

/usr/lib/passwd *:* Directorio principal.
/usr/lib/passwd/*.help *:* Archivos de ayuda
/usr/lib/passwd.*.motd *:* Mensajes que se pueden poner para la salida de comandos
/usr/lib/passwd/checkpassword *:* Programa que verifica si un password es débil o fuerte
/usr/lib/passwd/history_admin *:* Base de datos histórica
/usr/lib/passwd/npasswd *:* Programa a usar en lugar de passwd
/usr/lib/passwd/passwd.conf *:* Archivo de configuración
/usr/lib/passwd/bin : Herramientas de administración y para reinstalar o desinstalar npasswd
/usr/lib/passwd/dictionaries *:* Diccionarios procesados
/usr/lib/passwd/doc *:* Toda la documentación, incluyendo manuales y guías de administración

Después de la instalación, es necesario configurar el archivo :

```
/usr/lib/passwd/passwd.conf
```

se recomienda usar los siguientes parámetros :

```
MatchTries          3
MatchWait           2
PasswdTolerance      8
ShadowTolerance      32
passwd.AlphaOnly     no
passwd.CharClasses    4
passwd.Dictionaries   usr/lib/passwd/dictionaries
passwd.DisallowedChars '^C^S^Q^D^H^J^M^O^R^Y^Z^]\033^\\0177'
passwd.Help          /usr/lib/passwd/passwd.help
passwd.History       age     180
passwd.History       depth    4
passwd.History       database  dbm /usr/lib/passwd/history
passwd.LengthWarn     yes
passwd.MaxPassword    8
passwd.MaxRepeat      2
passwd.Message        /usr/lib/passwd/passwd.motd
passwd.MinPassword    6
passwd.PasswordChecks  lexical passwd local history dictionary
passwd.PrintableOnly  no
passwd.SingleCase     no
passwd.WhiteSpace     no
```

Después de toda la instalación se deben generar los diccionarios, como se muestra a continuación:

```
# cd  /usr/lib/passwd/bin
# ./makedict -o /home/fabian/software/npasswd/dict/Family-Names
# cd /home/fabian/software/npasswd/dict
```

se generan estos archivos :

```
-rw-r--r--   1 root    sys        16 Mar  8 18:52 Family-Names.pwi
-rw-r--r--   1 root    sys        16 Mar  8 18:52 Family-Names.pwd
-rw-r--r--   1 root    sys      1024 Mar  8 18:52 Family-Names.hwm
```

entonces se deben de ubicar correctamente los diccionarios; moverlos a /usr/lib/passwd/dictionaries

```
#mv Family* /usr/lib/passwd/dictionaries
```

Realizar los pasos anteriores con todos los diccionarios. Después verificar que funciona utilizando el comando checkpasswd :

```
# /usr/lib/passwd/checkpasswd
```

```
#./checkpassword Admin
```

```
Password to check: Admin
Password bad: it is too short (minimum length is 6 characters).

Password to check: Admin123
Password bad: needs more punctuation characters or whitespace.

Password to check: Admin@123
checkpassword: Only the first 8 characters of this password will be used
Password ok.
```

Al final indica que el password seleccionado se le considera fuerte.

La sintaxis para usar el npasswd es el siguiente :

```
# npasswd nombre_de_usuario
```

3.3 CIFRADO DE DATOS

3.3.1 PGP (Pretty Good Privacy)

Como se mencionó en el capítulo anterior, con PGP se pueden enviar mensajes de correo electrónico cifrados y también firmar o cifrar archivos personales. Todo esto con la finalidad de evitar que se altere información enviada e interceptada a través de la red, para esta tesis se vera la parte de cifrado de archivos, ya que utilizará en una red aislada de internet, por lo tanto, no se usará para los correos, sin embargo la funcionalidad es la misma.

A continuación se muestra la instalación de pgp, regularmente se encuentra en paquetes de instalación, lo que significa que no es necesario compilarlo, simplemente instalar el software :

La instalación de pgp consiste de lo siguiente :

```
# gunzip PGPcmdln_6.5.8_Sol_FW.tar.gz
# tar xvf PGPcmdln_6.5.8_Sol_FW.tar
```

En este caso no se compila absolutamente nada, solo se generan los binarios para su uso. Una vez que se haya instalado PGP, se puede empezar a intercambiar información de manera segura con otros usuarios de PGP. Obviamente es necesario tener las llaves públicas de ellos y distribuir la llave pública personal.

A continuación se muestra el uso de pgp, con las opciones más comunes y necesarias, para empezar a usarlo. Para obtener ayuda de cómo usar el pgp, se teclea:

```
# pgp –h
```

que nos muestra las opciones disponibles.

Lo primero que se debe hacer es generar las llaves para poder empezar a cifrar la información, esto se realiza con la opción –kg, que nos manda la siguiente salida :

```
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.
```

En este caso pregunta qué algoritmo se usará para generar la llave pública :

```
Choose the public-key algorithm to use with your new key
1) DSS/DH (a.k.a. DSA/ElGamal) (default)
2) RSA
Choose 1 or 2: 1
```

Después que tipo de llave va a generar, como es la primera ocasión que se genera, se escogería esa opción :

```
Choose the type of key you want to generate
1) Generate a new signing key (default)
2) Generate an encryption key for an existing signing key
Choose 1 or 2: 1
Pick your DSS ``master key" size:
1) 1024 bits- Maximum size (Recommended)
Choose 1 or enter desired number of bits: 1024
Generating a 1024-bit DSS key
```

También se requiere un UserID

```
You need a user ID for your public key. The desired form for this
user ID is your name, followed by your E-mail address enclosed in
<angle brackets>, if you have an E-mail address.
For example: John Q. Smith <jqsmith@nai.com>
Enter a user ID for your public key: Rosendo Fabian Hernandez <rfabian@tesofe.gob.mx>

Enter the validity period of your signing key in days from 0 - 10950
0 is forever (the default is 0):
```

Hay que proporcionar un password-frase para la llave, nos recomiendan usar una frase, son validos los espacios en blanco :

```
You need a pass phrase to protect your DSS secret key.
Your pass phrase can be any sentence or phrase and may have many
words, spaces, punctuation, or any other printable characters.

Enter pass phrase:
Enter same pass phrase again:

PGP will generate a signing key. Do you also require an
encryption key? (Y/n) Y
Pick your DH key size:
1) 1024 bits- High commercial grade, secure for many years
2) 2048 bits- "Military" grade, secure for forseeable future
3) 3072 bits- Archival grade, slow, highest security
Choose 1, 2, 3, or enter desired number of bits: 2

Enter the validity period of your encryption key in days from 0 - 10950
0 is forever (the default is 0):

Note that key generation is a lengthy process.
PGP needs to generate some random data. This is done by measuring
the time intervals between your keystrokes. Please enter some
random text on your keyboard until the indicator reaches 100%.
Press ^D to cancel
```

```
90% of
100% of required data
Enough, thank you.
...******* ..........................................******* .
Make this the default signing key? (Y/n) Y

Key generation completed.
```

Una vez completada la generación de la llave se procede a usar pgp. En el directorio HOME del usuario aparece un directorio llamado .pgp, es aquí donde estará el anillo de llaves que usa pgp. De este directorio debemos exportar nuestra llave pública a ASCII para poder brindársela a las demás personas que cuenten con pgp y pongan este archivo en su anillo de llaves :

```
# ./pgp -kxa "Rosendo Fabian Hernandez <fabo@servidor.unam.mx>"
```

```
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.

Extracting from keyring '/export/home/fabian/.pgp/pubring.pkr', userid "Rosendo Fabian
Hernandez < fabo@servidor.unam.mx >".

Extract the above key(s) into which file? fabo.asc

Output file 'fabo.asc' already exists. Overwrite (y/N)? y

Transport armor file: fabo.asc

Key extracted to file 'fabo.asc'.
```

El archivo generado tendrá algo similar a esto :

```
# more fabo.asc
```


```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGP 6.5.8

mQGiBDzS/FURBADr1dmNYtWWEXMxcYvB+pWEQsE1NW98f5w61ppU3wnfJY/kzMqA
hn7YNPD95fxAnazeJvZQkGdngn2Y5eX57qyTnR+F7XFNH4ayLyK8u5x9l62+VhKL
/fyC9pstbXFYB0SWB5Z6otfR+v2NF5a/roGvBlstt7Z/D8ScRzjkmoy0ewCg/1K6
PRJ0l4cfbjlYLHYq7ed09iMEAK46YBJLxzbhShjL2N7Ud3AMk6MG9+vkGkA9O1kT
CXrdzdbxsCxSW4yB6XtojBNG5HHdL1j06wygf+5giXjqXlB5s4677/Vs+Fdl4bkZ
Da9nohiQl6plyuQ7hYFoK5Xx4ggC8MpPFpKjYidsDHlHbp2ACcQhFiNZ61ufl+M9
wOB+BAC6JnE38z9l56brMVCrfgeKLioVyxCAoloixLc6Stw02o4Ldzhbq21M23tk
SDLKhgRRRP7sL1wp0woJFa6HUmZqOeVNGE/c5AdPYmN/B3mOoZU7MuUZx2/8allu
SAuo7LyisbozJWO54M909HSDjKAnwEQVBUTLHX6S3ErhtSVzEydyhxO5sfFAfjO1
uL/oarze7mnnSWG565HsB5NwCaNrCgs2gWpwHld+1Aa6CUjg8tXXdxVZdL0OEi4a
SqrWbSMrliNd4vBYaArYTBr95rGx5uXump8rsvzN8tuRxBfn4bTsB822jl02fU7U
q2gfjf5RIFKCpNArmRitbBgcEpF0OFIYHi1fJXBepqE4wAmHhjTuY41vDAZu0AJW
l1SeczNu1SAYdSWUZ0lK6Wahkobgyvc1iQBGBBgRAgAGBQl80vxfAAoJEAE5dwEX
b2lHbjsAn1v1UX8UvCqubH0Baw4sd5CtMNldAKDAd4W62FmelgVZkDsZajtPApVf
AA==
=V3um
-----END PGP PUBLIC KEY BLOCK-----
```


Ahora ya podemos empezar a distribuir archivos cifrados, de la siguiente manera se puede cifrar un archivo :

```
# pgp -ca nombre_del_archivo
# pgp -ca prueba.txt
```

```
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.

You need a pass phrase to encrypt the file.
```

```
Enter pass phrase:
Enter same pass phrase again:
```

Aquí se le da el password (frase que se tecleo al generar las llaves)

```
Transport armor file: prueba.txt.asc
```

Y genera el archivo prueba.txt.asc, el cual estará cifrado. El contenido de ambos archivos estarán así :

 # more prueba.txt

```
Network Associates Freeware End User License Agreement
(Non-Commercial Use and Distribution Only)
NOTICE TO ALL USERS: CAREFULLY READ THE FOLLOWING LEGAL
AGREEMENT ( "AGREEMENT"), FOR THE LICENSE OF SPECIFIED SOFTWARE
("SOFTWARE") BY NETWORK ASSOCIATES TECHNOLOGY COMPANY, INC.
```

 # more prueba.txt.asc


```
-----BEGIN PGP MESSAGE-----
Version: PGP 6.5.8

pRNM6XLmuiTwYzcc/L1EZsVGiAkPK1lE0YV7WY7eS6ARZrJN4sAAxkMtF3ha+iON
kHy7cTbRAfLbIFRIFvyGtRSbg9e3QWKoifqqpwu80VSbVpY+T37PaMcllyMG9V4g
BNLU1sahJGzc6572R1SVw42eXJplB5peB8i79loQF5ASKWooJRYepHS9YZ4VdY+/
tmj8F9oZVGVw1k3tzRLRwX5GXWhlX4/uzellwm/me042aCSurdZZDl8Vgz3ZjxZR
yPwmA33HmJp6EDGVYBdtRVrljCw9qx9xeicW22yP3kmbQMJDWHFSy3QH5UpPIOXA
PGkUoB/R4sStADgfL05Dmttj9fvp98MhOn2q5bhU6A5Zxs1rSRgJfVkqeHLLoTnx
J7KoJkm6V1GerCH/BKt6WwpsbevJN5Y9lllXLl26DvuUEalfl1LiMQcXZQVWkKf7
gooFPeosnRwrFoW76LbYCjoLcEDhXHUXeA5hNb6KXj/NUGs8ZdwnrH0pTe7kny8i
jNC8u9N2nN1oae2U7qNETAxFL1M5BgQpAA71l4Z54m8Jf8PbpAyBPvjykARX+cyH
/3MeMhoGEAZ3kec8RmOgiivcnj3ag7fUmUSnEfPe2jtg7pgS0bj6JVU1/RVYUVGd
G6OQG1+Wry03px7coF2sdAGzGa3ObL92CCbXoCL0luurTEDaUL5v85rzesu3A+d3
MvqVe9Jj/VM7XCFOL8plBl1zwBGspkTLPVG4PQkJHCKpol+9GOidWcrAl6xk97CB
```


Este archivo ya se puede enviar a las personas indicadas para que lo puedan visualizar. De manera general éste es el uso más frecuente que se le da a PGP.

3.4 COMUNICACIÓN

3.4.1 Secure shell

La instalación de ssh consiste de lo siguiente :

```
# gunzip ssh-1.2.31.tar.gz
# tar xvf ssh-1.2.31.tar
```

Correr el script configure, el cual es un shell que determinará el ambiente del sistema para poder construir ssh. Definirá que clase de sistema está corriendo, características y tipo de compilador.

 # ./configure

Después de haber armado la configuración, se procede a la compilación del software con el comando make :

 # make

El siguiente paso es la instalación de secure shell :

 # make install

Con esto se generan los archivos de configuración :

```
/etc/ssh_host_key y /etc/sshd_config
```

los programas clientes :

```
/usr/local/bin/ssh y /usr/local/bin/scp
```

El demonio de secure shell está en :

```
/usr/local/sbin/sshd.
```

Ahora se tiene que habilitar el puerto 22, que es por donde se harán las peticiones, en los archivos /etc/inetd.conf y /etc/services :

En el archivo /etc/inetd.conf insertar la siguiente línea, que a continuación se marca en negritas :

```
#       ARPA/Berkeley services
#
##
ftp       stream tcp nowait root /usr/lbin/ftpd     ftpd -l
telnet     stream tcp nowait root /usr/lbin/telnetd  telnetd
ssh    stream  tcp    nowait  root   /usr/lbin/sshd   sshd -i
tftp      dgram  udp wait   root /usr/lbin/tftpd   tftpd\
#bootps     dgram  udp wait  root /usr/lbin/bootpd  bootpd
#login      stream tcp nowait root /usr/lbin/rlogind  rlogind
```

y en el archivo /etc/services insertar las siguientes líneas, que a continuación se marcan en negritas :

```
systat     11/tcp  users          # Active Users
daytime     13/tcp               # Daytime
daytime     13/udp               #
qotd       17/tcp  quote          # Quote of the Day
chargen     19/tcp  ttytst source  # Character Generator
chargen     19/udp  ttytst source  #
ftp-data    20/tcp               # File Transfer Protocol (Data)
ftp        21/tcp               # File Transfer Protocol (Control)
ssh        22/tcp  Secure Shell
ssh        22/udp  Secure Shell
telnet      23/tcp               # Virtual Terminal Protocol
smtp        25/tcp               # Simple Mail Transfer Protocol
time        37/tcp  timeserver    # Time
time        37/udp  timeserver    #
rlp        39/udp  resource      # Resource Location Protocol
whois       43/tcp  nicname       # Who Is
domain      53/tcp  nameserver    # Domain Name Service
domain      53/udp  nameserver    #
```

Después de hacer estas respectivas modificaciones, hay que levantar el demonio de **inetd** para que identifique que ya existe otro puerto que estará recibiendo peticiones :

```
#ps -fea | grep inetd  → para obtener el ProcessID (PID)

root  491    1  0  Dec 20  ?      0:00 /usr/sbin/inetd
```

Después de obtener el PID, hay que levantar y tirar el demonio :

 # kill –HUP 491 → PID

En este momento ya está instalado secure shell, ahora se instala el cliente en cada PC que se quiera tener como cliente. Estos archivos se pueden obtener en internet, puesto que también son de libre distribución:

Instalación de secure shell y secure copy para windows

1) Descompactar los archivos :

ssh3297332.zip
crypl110.zip
patch01.zip
ssh-dos.zip

en el directorio c:\ssh

2) Activar la opción de mostrar todos los archivos (incluyendo ocultos) en el explorador de windows

3) Correr el archivo del parche:

To patch DES in cryptlib 1.1 simply run
PATCH CRYPT32.PAT CRYPT32.DLL

Crear acceso directo de secure shell

4) Para poder utilizar el scp :

a) Crea el directorio c:\ssh\etc
b) Establece tu variable de ambiente en el archivo c:\autoexec.bat
set PATH=c:\ssh
Activar el autoexec.bat (corriéndolo de nuevo)
Establece tu variable de ambiente HOME: set HOME =c:\ssh
c) Crea el directorio c:\ssh\.ssh

5) Como utilizar scp:

a) De la máquina local (pc) al servidor:

scp nombre_archivo login@hostname:ruta_del_destino

los dos puntos **(:)** después del hostname, indican la ruta del HOME

ejemplo:

scp c:\windows\prueba.txt fabo@servidor.unam.mx:

b) Del Servidor a la pc:

scp login@hostname:nombre_archivo ruta_del_destino

ejemplo:

scp fabo@servidor.unam.mx:hola.txt c:\windows

6) Verificar si los archivos están en los destinatarios, dentro de la cuenta del servidor o dentro de la carpeta en la PC.

3.4.2 OpenSSH

Como se mencionó en el capítulo anterior, el openssh tiene las mismas funciones que secure shell, la diferencia que éste es una versión libre, la cual es compatible con los protocolos SSH1 y SSH2. El openssh también cuenta con el demonio sshd y funciones del secure shell, tales como ssh-add, ssh-agent y ssh-keygen.

openSSH corre en las siguientes plataformas :

```
OpenBSD
Linux
Solaris
AIX
IRIX
HP/UX
FreeBSD
NetBSD
```

El openSSH requiere la instalación del siguiente software que también es de libre distribución : zlib y OpenSSL

zlib

Es necesario instalar este software ya que cuenta con librerías necesarias para el funcionamiento de SSH y OpenSSH.

La instalación de zlib consiste de lo siguiente :

```
# gunzip zlib.tar.gz
# tar -xvf zlib.tar
```

Correr el script configure, el cual es un shell que determinará el ambiente del sistema para poder construir zlib.

```
# ./configure
```

Después de haber armado al configuración, se procede a la compilación del software con el comando make :

```
# make
```

El siguiente paso es la instalación de zlib :

```
# make install
```

OpenSSL

La instalación de openssl consiste de lo siguiente :

```
# gunzip openssl-0.9.6.tar.gz
# tar -xvf openssl.0.9.6.tar
```

Correr el script config, el cual es un shell que determinará la plataforma del sistema.

```
# ./config
```

Con lo anterior ya se puede realizar la configuración de la herramienta :

 # ./configure "plataforma" → obtenida en el paso anterior

Después de haber armado la configuración, se procede a la compilación del software con el comando make :

```
# make
```

El siguiente paso es la instalación de openssl :

```
# make install
```

OpenSSH

La instalación de openssh consiste de lo siguiente :

```
# gunzip openssh-x.x.x.tar.gz
# tar -xvf openssh-x.x.x.tar
```

Correr el script configure, el cual es un shell que determinará el ambiente del sistema para poder construir openssh.

```
# ./configure
```

Después de haber armado la configuración, se procede a la compilación del software con el comando make :

```
# make
```

El siguiente paso es la instalación de openssh :

```
# make install
```

Con todo lo anterior ya se tiene instalado el producto openssh, para revisar que se instaló correctamente, obviamente si no se reportaron errores durante la instalación, se teclean los siguientes comandos :

```
# ssh –v
```

```
ssh: SSH Secure Shell 3.0.1
Compiled with SSL (0x0090600f).
```

Para poder empezar a utilizar el openssh, hay que levantar el demonio de secure shell, de la siguiente manera :

```
# /usr/lbin/sshd
```

en este caso se encuentra en esta ruta.

Por último se deben de generar las llaves de cifrado que ocupará openssh :

```
# ssh-keygen -t rsa1 -f /etc/ssh/ssh_host_key -N ""
# ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key -N ""
# ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key -N ""
```

Las rutas pueden cambiar dependiendo del lugar de instalación de los archivos.

3.5 MONITOREO DE RED

3.5.1 TCP-wrappers

Como se señaló en el capítulo anterior TCP-Wrappers nos servirá como una protección a los servicios de red que brinda el servidor, a continuación se muestra la instalación de esta herramienta y algunos "tips" de configuración.

La instalación de tcp-wrappers consiste de lo siguiente :

```
# gunzip tcp_wrappers_7.6.tar.gz
# tar xvf tcp_wrappers_7.6.tar
```

Antes de empezar la compilación se deberán realizar algunos ajustes al archivo Makefile que se encuentra debajo del directorio que fue creado al desempacar el tar de tcp-wrappers.

Se edita el archivo y se modifican las siguientes líneas :

```
REAL_DAEMON_DIR=/usr/sbin
```

Esta variable indica la ruta donde se encuentran los demonios de los servicios de red, en este caso se encuentran debajo de /usr/sbin, esta ruta se muestra en el archivo /etc/inetd.conf :

```
ftp    stream tcp   nowait root  /usr/sbin/in.ftpd     in.ftpd
telnet stream tcp   nowait root  /usr/sbin/in.telnetd  in.telnetd
```

la otra línea a modificar es :

```
FACILITY=LOG_MAIL        por
FACILITY=LOG_LOCAL0
```

Aquí se le indica que los mensajes que genere tcp-wrappers se registren de manera independiente que los de sendmail y por último se le quita el comentario a esta línea:

```
STYLE = -DPROCESS_OPTIONS
```

para activar las opciones de banner.

Para poder compilar la herramienta, es necesario definir qué compilador se usará, esta opción se habilita en la sección que le corresponda al sistema operativo en donde será instalado, por ejemplo, si se va a instalar sobre una plataforma de sunOS, entonces se modificará la parte marcada con negritas :

```
# SunOS 5.x is another SYSV4 variant.
sunos5:
    @make REAL_DAEMON_DIR=$(REAL_DAEMON_DIR) STYLE=$(STYLE) \
    LIBS="-lsocket -lnsl" RANLIB=echo ARFLAGS=rv VSYSLOG= \
    NETGROUP=-DNETGROUP AUX_OBJ=setenv.o TLI=-DTLI CC=gcc \
    BUGS="$(BUGS) -DSOLARIS_24_GETHOSTBYNAME_BUG" all
```

Después de haber realizado todos esos cambios en la configuración, se procede a la compilación del software con el comando make y el nombre de la plataforma, en donde se instalará, este último dato se puede obtener tecleando solo make, lo que desplegará una lista de plataformas, como a continuación se muestra :

```
# make
```

```
This Makefile knows about the following sys-types:
    generic (most bsd-ish systems with sys5 compatibility)
    386bsd aix alpha apollo bsdos convex-ultranet dell-gcc dgux dgux543
    dynix epix esix freebsd hpux irix4 irix5 irix6 isc iunix
    linux machten mips(untested) ncrsvr4 netbsd next osf power_unix_211
    ptx-2.x ptx-generic pyramid sco sco-nis sco-od2 sco-os5 sinix sunos4
    sunos40 sunos5 sysv4 tandem ultrix unicos7 unicos8 unixware1 unixware2
    uts215 uxp
```

En este ejemplo se ocupó sunos5, entonces se teclearía de la siguiente manera:

```
# make sunos5
```

Una vez que se compiló correctamente, se realizan los siguientes cambios al sistema :

En el directorio actual se generan los archivos : tcpd, tcpdmatch, tcpdchk, try_from y safe_finger, estos deberán moverse al directorio en donde se encuentran los demonios de red, como se vio en párrafos anteriores.

```
# mv tcpd tcpdmatch tcpdchk try-from safe_finger /usr/sbin
```

Modificar en el archivo /etc/inetd.conf la ruta de los demonios, por lo siguiente, /usr/sbin/tcpd, quedará de la siguiente manera :

```
ftp    stream tcp   nowait root  /usr/sbin/tcpd      in.ftpd
telnet stream tcp   nowait root  /usr/sbin/tcpd   in.telnetd
ssh    stream tcp   nowait root  /usr/sbin/tcpd      sshd –I
```

Añadir la siguiente línea en el archivo /etc/syslog.conf para indicarle en donde se localizara la bitácora de tcp-wrappers :

```
local0.info                    /usr/local/adm/tcpd.log
```

Este archivo deberá crearse y cambiarle los permisos para que solo root pueda leerlo.

```
# touch /usr/local/adm/tcpd.log
# chmod 600 /usr/local/adm/tcpd.log
```

Después de estos cambios, se tendrán que tirar y levantar los demonios de inetd y syslogd

kill -HUP PID → Proccess ID de inetd y syslogd

Verificar que se registren los accesos en la bitácora de tcpd, haciendo pruebas de conexión con cualquiera de los servicios de red, por ejemplo telnet y ftp:

```
# telnet localhost
# ftp localhost
```

En la bitácora deberá aparecer algo así :

```
Mar 21 07:37:27 tfsun1 in.telnetd[11983]: connect from 10.150.37.72
Mar 21 07:37:41 tfsun1 in.ftpd[11985]: connect from 10.150.37.72
Mar 21 07:37:42 tfsun1 in.telnetd[11986]: connect from 10.150.37.72
Mar 21 07:37:56 tfsun1 in.ftpd[11988]: connect from 10.150.37.72
```

Tcp-wrappers tiene muchas opciones de configuración, solo se verán las mas relevantes :

Existen 2 archivos /etc/hosts.allow y el /etc/hosts.deny, en los que se puede definir que equipos deseamos que tengan permiso para utilizar los servicios de red y que equipos no, la sintaxis es la siguiente :

```
servicio que se otorgará : equipos permitidos o denegados : ejecuciones opcionales
```

Por ejemplo:

En el archivo /etc/hosts.allow

```
in.ftpd, in.telnetd : 10.150.32.32
```

Indica que le otorga el acceso al equipo 10.150.32.32 al servicio de ftp y telnet. Para denegar los servicios restantes a todos los demás equipos :

En el archivo /etc/hosts.deny

```
ALL:ALL
```

También se podrían definir rangos de acceso al servidor :

```
ALL:10.150.80.1/10.150.80.23
```

Uno de los usos más interesantes es el que se muestra en el siguiente ejemplo :

```
ALL:ALL: (/usr/sbin/safe_finger -l @%h | /bin/mail root)
```

Esta línea del /etc/hosts.deny indica que se le niega el acceso de todos los servicios a todos los equipos, manda un finger (propio de tcp-wrappers) al equipo que intentó realizar la conexión y a su vez manda un correo, en este caso al usuario root, avisándole que equipo intentó conectarse.

El servidor primero lee el archivo /etc/hosts.allow y después el /etc/hosts.deny, entonces hay que tener cuidado en como definir las reglas, ya que se podrían contradecir las de un archivo con las del otro.

3.5.2 sniffer (como herramienta analizadora)

Como se había mencionado toda la información que viaja por la red se transmite en claro. El sniffer que se utilizará en esta tesis será el que viene incluido con Solaris. Se llama snoop y es un programa que puede monitorear y analizar el tráfico de la red. Lo primero que se debe verificar es que este comando solo lo pueda ejecutar el usuario root.

Es muy difícil encontrar sniffers de libre distribución (excepto para linux), pero en este caso se aprovechará una herramienta propia del sistema operativo, en el caso de HP-UX el comando que se utiliza es nettl, que es muy similar al snoop de Solaris.

Solo se verán algunas opciones del snoop, aplicándose en ejemplos reales :

Por ejemplo :

```
# snoop -o info port 23 99.150.6.205 10.150.37.180
```

En este caso , captura paquetes entre 2 equipos que se comuniquen por el puerto 23 y la información la guarda en el archivo info

Para analizar el archivo info :

```
# snoop -i info -t r
```

mandara un contenido similar a éste :

```
10  1.31657 10.150.37.180 -> tfsun6    TELNET C port=4498 f
11  1.31703 10.150.37.180 -> tfsun6    TELNET C port=4498 a
12  1.33908 10.150.37.180 -> tfsun6    TELNET C port=4498 b
13  1.37860 10.150.37.180 -> tfsun6    TELNET C port=4498 i
14  1.47203 10.150.37.180 -> tfsun6    TELNET C port=4498 a
15  1.55944 10.150.37.180 -> tfsun6    TELNET C port=4498 n
16  1.70932 10.150.37.180 -> tfsun6    TELNET C port=4498
20  2.59162 10.150.37.180 -> tfsun6    TELNET C port=4498 f
21  2.67024 10.150.37.180 -> tfsun6    TELNET C port=4498 a
22  2.83222 10.150.37.180 -> tfsun6    TELNET C port=4498 b
23  2.87937 10.150.37.180 -> tfsun6    TELNET C port=4498 i
24  2.99266 10.150.37.180 -> tfsun6    TELNET C port=4498 a
25  3.07334 10.150.37.180 -> tfsun6    TELNET C port=4498 n
26  3.31357 10.150.37.180 -> tfsun6    TELNET C port=4498 1
27  3.35928 10.150.37.180 -> tfsun6    TELNET C port=4498 2
28  3.55423 10.150.37.180 -> tfsun6    TELNET C port=4498 3
29  3.75287 10.150.37.180 -> tfsun6    TELNET C port=4498 4
32  4.21279 10.150.37.180 -> tfsun6    TELNET C port=4498
33  6.70838 10.150.37.180 -> tfsun6    TELNET C port=4498 p
35  6.82862 10.150.37.180 -> tfsun6    TELNET C port=4498 w
36  6.86821 10.150.37.180 -> tfsun6    TELNET C port=4498 d
39  9.74957 10.150.37.180 -> tfsun6    TELNET C port=4498
40  9.62928 10.150.37.180 -> tfsun6    TELNET C port=4498 l
41  9.74957 10.150.37.180 -> tfsun6    TELNET C port=4498 s
42  9.86987 10.150.37.180 -> tfsun6    TELNET C port=4498
45  11.13022 10.150.37.180 -> tfsun6   TELNET C port=4498 e
47  11.32870 10.150.37.180 -> tfsun6   TELNET C port=4498 x
49  11.52736 10.150.37.180 -> tfsun6   TELNET C port=4498 i
50  11.64772 10.150.37.180 -> tfsun6   TELNET C port=4498 t
51  11.82335 10.150.37.180 -> tfsun6   TELNET C port=4498
```

Aquí podemos notar que manda información relevante como los puertos ocupados, el servicio solicitado, lo tecleado desde el equipo cliente (como se puede ver en la última columna, que se tecleo el login, el password y un par de comandos) y la información que entrega el servidor. Podemos usar esta información con muchos fines, por ejemplo, si alguien se está conectando indebidamente, que comandos está ejecutando y que está realizando. Sin embargo, el uso más adecuado es cuando se tienen problemas de conexión en algunas aplicaciones y el problema radica en que los paquetes no están llegando correctamente. El comando snoop nos ayudará a ver que está pasando por la red a nivel de paquetes y analizar cual es el problema.

Para analizar más a detalle un paquete capturado en un archivo :

 # snoop -i info -v -p10

y mandará una salida similar a ésta :

 # snoop -i info -v -p10

```
ETHER: Packet 10 arrived at 20:13:44.33
ETHER: Packet size = 60 bytes
ETHER: Destination = 8:0:00:c9:4f:cc, Sun
ETHER: Source    = 0:80:2d:bf:3e:9, Xylogics, Inc.  Annex terminal servers
ETHER: Ethertype = 0800 (IP)
ETHER:
IP: ----- IP Header -----
IP: Version = 4
IP: Header length = 20 bytes
IP: Type of service = 0x00
IP:     xxx. .... = 0 (precedence)
IP:     ...0 .... = normal delay
IP:     .... 0... = normal throughput
IP:     .... .0.. = normal reliability
IP: Total length = 41 bytes
IP: Identification = 46952
IP: Header checksum = acad
IP: Source address = 10.150.37.180, 10.150.37.180
IP: Destination address = 99.150.6.205, pinky
IP: No options
IP:
TCP: ----- TCP Header -----
TCP:
TCP: Source port = 4498
TCP: Destination port = 23 (TELNET)
TCP: Sequence number = 3297294597
TCP: Acknowledgement number = 2297251623
TCP: Data offset = 20 bytes
TCP: Flags = 0x18
TCP:
TELNET: ----- TELNET: -----
TELNET: "f"
```

Con estas opciones se pueden verificar los encabezados del paquete, en donde viene información más completa y detallada de los paquetes analizados.

Otra opción que proporcionan información es el siguiente, solo que éste lo va mostrando en pantalla y todos los paquetes, no monitorea un puerto en especial, si no todo lo que pase por la tarjeta :

 # snoop -d interface IP_servidor
 # snoop –d hme0 99.150.6.205

```
Using device /dev/hme (promiscuous mode)
   pinky -> 10.150.37.180 TCP D=1451 S=22   Ack=2810908036 Seq=3767268879 Len=40
10.150.37.180 -> pinky    TCP D=22 S=1451   Ack=3767268919 Seq=2810908036 Len=0
10.150.37.72 -> pinky     TELNET C port=4325
   pinky -> 10.150.37.72 TELNET R port=4325
   pinky -> 10.150.37.72 TELNET R port=4325
10.150.37.72 -> pinky     TELNET C port=4325
10.150.37.72 -> pinky     HTTP C port=4331
   pinky -> 10.150.37.72 HTTP R port=4331
10.150.37.72 -> pinky     HTTP C port=4331
10.150.37.72 -> pinky     HTTP HEAD / HTTP/1.0
   pinky -> 10.150.37.72 HTTP R port=4331
10.150.37.72 -> pinky     UDP D=161 S=161 LEN=47
   pinky -> 10.150.37.72 UDP D=161 S=161 LEN=69
   pinky -> 10.150.37.180 TCP D=1451 S=22   Ack=2810908068 Seq=3767269159 Len=48
10.150.37.180 -> pinky    TCP D=22 S=1451   Ack=3767269207 Seq=2810908068 Len=0
   pinky -> 10.150.37.180 TCP D=1451 S=22   Ack=2810908068 Seq=3767269207 Len=48
```

```
10.150.37.180 -> pinky     TCP D=22 S=1451     Ack=3767269447 Seq=2810908068 Len=0
10.150.37.72 -> pinky      FTP C port=4332
10.150.37.72 -> pinky      FTP C port=4332
7     pinky -> 10.150.37.72 FTP R port=4332 220 pinky FTP serve
10.150.37.72 -> pinky      FTP C port=4332 QUIT\r\n
   pinky -> 10.150.37.72 FTP R port=4332
   pinky -> 10.150.37.72 FTP R port=4332 221 Goodbye.\r\n
   pinky -> 10.150.37.72 FTP R port=4332
10.150.37.72 -> pinky      FTP C port=4332
10.150.37.72 -> pinky      FTP C port=4332
   pinky -> 10.150.37.72 FTP R port=4332
10.150.37.72 -> pinky      TELNET C port=4338
   pinky -> 10.150.37.72 TELNET R port=4338
10.150.37.72 -> pinky      TELNET C port=4338
   pinky -> 10.150.37.180 TCP D=1451 S=22    Ack=2810908068 Seq=3767269447 Len=80
   pinky -> 10.150.37.180 TCP D=1451 S=22    Ack=2810908068 Seq=3767269527 Len=64
10.150.37.72 -> pinky      HTTP C port=4342
   pinky -> 10.150.37.72 HTTP R port=4342
10.150.37.72 -> pinky      UDP D=161 S=161 LEN=47
   pinky -> 10.150.37.72 UDP D=161 S=161 LEN=69
   pinky -> 10.150.37.180 TCP D=1451 S=22    Ack=2810908068 Seq=3767269903 Len=72
```

Aquí se ve todo lo que está pasando por la tarjeta hme0 del servidor.

3.5.3 Portsentry

Esta herramienta monitorea barridos a los puertos de un servidor, se marcan en una bitácora, se bloquea la comunicación con la máquina que está realizando el scanneo y se ejecuta una acción (algún comando), especificada por el administrador.
La instalación de portsentry consiste de lo siguiente :

```
# gunzip portsentry-1.0.tar.gz
# tar xvf portsentry-1.0.tar
```

Antes de realizar la compilación se deben hacer los siguientes cambios al archivo Makefile :

línea 26 : Definir que compilador se usará cc ó gcc
CC = gcc
línea 38 : Definir en que directorio se instalará portsentry
INSTALLDIR = /usr/local/psionic

Después de haber realizado estos cambios en la configuración, se procede a la compilación del software con el comando make :

```
# make
```

que nos arroja la siguiente salida :

```
Usage: make <systype>
<systype> is one of: linux, bsd, solaris, hpux, hpux-gcc,
freebsd, openbsd, netbsd, bsdi, aix, osf, generic
directory: /usr/local/psionic
```

Esto indica que debe ejecutarse : make <tipo de sistema>, para este ejemplo es de la siguiente manera :

```
# make solaris
```

```
SYSTYPE=solaris
Making
gcc -lnsl -lsocket -lc -o ./portsentry ./portsentry.c ./portsentry_io.c \
    ./portsentry_util.c
```

El siguiente paso es la instalación de portsentry :

```
# make install
```

```
Creating psionic directory /usr/local/psionic
Setting directory permissions
chmod 700 /usr/local/psionic
Creating portsentry directory /usr/local/psionic/portsentry
Setting directory permissions
chmod 700 /usr/local/psionic/portsentry
Copying files
cp ./portsentry.conf /usr/local/psionic/portsentry
cp ./portsentry.ignore /usr/local/psionic/portsentry
cp ./portsentry /usr/local/psionic/portsentry
Setting permissions
chmod 600 /usr/local/psionic/portsentry/portsentry.ignore
chmod 600 /usr/local/psionic/portsentry/portsentry.conf
chmod 700 /usr/local/psionic/portsentry/portsentry
Edit /usr/local/psionic/portsentry/portsentry.conf and change
your settings if you haven't already. (route, etc)
```

La salida muestra todo el proceso de instalación que realiza y al final nos muestra la ubicación del archivo de configuración para la correcta ejecución de portsentry. Para el propósito de esta tesis se realizarán cambios básicos en el archivo de configuración (portsentry.conf), los cuales se mencionan a continuación:

1 : Quitar los símbolos de comentario a las especificaciones de los puertos (TCP y UDP) que sean poco solicitados, los cuales son especificados en las opciones UDP_PORTS y TCP_ PORTS

```
TCP_PORTS="1,11,15,110,111,143,540,635,1080,524,2000,12345,12346,20034,32771,32772,32773,32774,49724,54
320"
UDP_PORTS="1,7,9,69,161,162,513,640,700,32770,32771,32772,32773,32774,31337,54321"
```

2 : Para este caso, como ya se tiene instalado TCP-Wrappers, es conveniente quitar el símbolo de comentario a la siguiente línea :

```
KILL_HOSTS_DENY="ALL: $TARGET$"
```

3 : Se puede ejecutar una acción (algún comando), en la siguiente línea :

```
KILL_RUN_CMD="comando_o_script_a_ejecutar"
```

Con esto el servidor se encarga de monitorear si están haciendo un barrido de puertos y tomar las medidas necesarias. En el directorio /usr/local/psionic/portsentry, se alojan las bitácoras.

3.6 MONITOREO DEL SISTEMA

3.6.1 Cops

Como se mencionó anteriormente ésta es una herramienta para detectar vulnerabilidades dentro de nuestro sistema, la instalación es sencilla, solo que a diferencia de las anteriores se deben de realizar algunos ajustes y algunos "trucos" para evitar que surjan posibles errores al momento de instalarla.

La instalación de cops consiste de lo siguiente:

```
# gunzip cops.1.04.tar.gz
# tar xvf cops.1.04.tar
```

Antes de empezar la compilación se deberán realizar algunos ajustes, primero se obtienen rutas de dependencia entre archivos y compiladores, ejecutando el siguiente script, que se encuentra en el directorio generado al momento de descompactar el archivo tar :

```
# ./reconfig
```

después es necesario realizar algunos cambios en el archivo cops, ubicado en la misma ruta. Se edita el archivo :

```
# vi cops
```

y se modifican las siguientes líneas :

línea 70 : En esta línea se activa el módulo suid.chk, poniendo esta variable en yes :

```
RUN_SUID=YES
```

Y quitar el símbolo de comentario también las líneas 230 – 232

```
if $TEST "$RUN_SUID" = "YES" ; then
    $SECURE/suid.chk > /dev/null 2>&1
fi
```

línea 93 : Aquí se define la ruta de los módulos de cops, en la variable SECURE que por default trae :

```
SECURE=/usr/foo/bar
```

Y debe de quedar la ruta en donde se bajo el software de cops, por ejemplo:

```
SECURE=/opt/software/cops_104
```

línea 94 : Aquí se define el correo electrónico del usuario que ejecutara cops, en este caso se escogió al usuario root :

```
SECURE_USERS=root@localhost
```

línea 192 : En esta línea esta la rutina que verifica vulnerabilidades al ftp anónimo, en caso de que exista éste en el servidor colocar la siguiente bandera (-a) :

```
if $TEST -n "$verbose" ; then
    $ECHO "**** ftp.chk ****" >> $VERBUCKET ; fi
$SECURE/ftp.chk  -a  >>    $RESULT 2>> $BIT_BUCKET
```

línea 198 : Aquí se le ponen comentarios al módulo de pass.chk (solo si nuestro servidor tiene habilitado el /etc/shadow) debe de quedar de la siguiente manera :

```
#if $TEST -n "$verbose" ; then
#$ECHO "**** pass.chk ****" >> $VERBUCKET ; fi
#$SECURE/pass.chk -w ./pass.words -b -g -s -c -d -n >> $RESULT 2>> $BIT_BUCKET
# $SECURE/pass_diff.chk      >>     $RESULT 2>> $BIT_BUCKET
```

línea 218 : Aquí está el módulo crc.chk y esta parte viene deshabilitada por default. Es la que se encarga de verificar el estado de los binarios, se deben de quitar los comentarios, de tal manera que quede de la siguiente forma :

```
if $TEST -n "$verbose" ; then
     $ECHO "**** crc.chk ****" >> $VERBUCKET ; fi
 $SECURE/crc.chk        2>> $BIT_BUCKET
#crc.chk puts it's results in a file called crc.results; uncomment
 #this as well:
 if $TEST -s "$SECURE/crc_results" ; then
     $CAT $SECURE/crc_results >> $RESULT
 fi
```

Después se tiene que modificar el tipo de compilador que se está usando, en la línea 41 del archivo makefile, en este caso quedó :

```
CC=/usr/local/bin/gcc
```

Después de haber realizado todos esos cambios en la configuración, se procede a la compilación del software con el comando make all

```
# make all
```

Una vez que se instaló de manera correcta, esto es, que no se hayan generado errores, se podrá usar cops de la siguiente manera :

```
# cops –v
```

Este comando va a generar un archivo, con una lista de las posibles vulnerabilidades que presente el sistema. Este archivo estará debajo de un directorio que tendrá el nombre del equipo y a su vez el archivo tendrá de nombre la fecha en que se generó, con el formato año_mes_dia, por ejemplo :

```
$COPS/libra/2002_Apr_2
```

que tendrá más o menos el siguiente contenido :

```
ATTENTION:
Security Report for Fri Apr 02 18:02:29 CDT 2002
from host libra
**** root.chk ****
Warning! Root does not own the following file(s):
/.profile
**** dev.chk ****
**** is_able.chk ****
Warning! /etc/security is _World_ readable!
Warning! /etc/SnmpAgent.d is _World_ writable!
Warning! /etc/rc.config.d is _World_ writable!
Warning! /usr/adm/snmpd.log is _World_ writable!
Warning! /usr/adm/spellhist is _World_ writable!
**** rc.chk ****
Warning! File /etc/rc.config.d/SnmpTrpDst (in /etc/rc.config) is _World_ writable!
Warning! File /opt/OV/bin/nettl (in /etc/rc3.d/S98netmgt) is _World_ writable!
Warning! File /opt/OV/bin/ovstart (in /etc/rc3.d/S98netmgt) is _World_ writable!
*** cron.chk ****

**** home.chk ****
Warning! User nuucp's home directory /var/spool/uucppublic is mode 01777!
**** passwd.chk ****
Warning! Password file, line 23, user respaldo has uid = 0 and is not root
     respaldo:x:0:3:Cuenta para respaldos:/export/home/respaldo:/bin/ksh
**** user.chk ****
**** misc.chk ****
```

```
**** ftp.chk ****
Warning! /etc/ftpusers should exist!
**** kuang ****
**** crc.chk ****
**** bug.chk ****
Warning! /usr/lib/sendmail could have a hole/bug! (CA-88:01)
Warning! /usr/lib/sendmail could have a hole/bug! (CA-90:01)
Warning! /bin/mail could have a hole/bug! (CA-91:01a)
```

Después de verificar este archivo, se procede a arreglar lo que consideremos sea una vulnerabilidad.

3.6.2 Tripwire

En el capítulo anterior se mencionó que tripwire es una herramienta que verifica la integridad de un sistema Unix, en: la estructura de archivos, filesystems, cambios en permisos de archivos, ligas, modificación de archivos críticos, UID's y GID´s, etc.

La instalación de tripwire consiste de lo siguiente:

```
# gunzip tripwire-1.2.tar.gz
# tar xvf tripwire-1.2.tar
```

Antes de empezar la compilación se deberán realizar ajustes en algunos archivos :

1) Primero se edita el archivo Ported, que se encuentra debajo del directorio generado al momento de descompactar el archivo tar :

```
# vi Ported
```

En este archivo se encuentran definidos los parámetros de todas las plataformas, se debe buscar la plataforma que nos interesa y tomar los datos que ahí aparecen, por ejemplo, si nuestra plataforma es hp-ux, buscamos la cadena hp-ux :

```
vendor:      HP
os:          HP/UX
os version:  8.x, 9.x
compiler:    cc
cflags:      -O -Aa -N          (ansi)
cflags:      -O -Ak -N          (k&r)
cflags:      -O -Wl,-a,archive -O -Ac (ensure archived, NO shared libraries)
conf.h:      conf-hpux.h
notes:       from Lance Bailey:
notes:            -Aa           ansi
notes:            -Ac           K&R
notes:         -Wl,-a,archive   ensure archived and NOT shared libraries on linking
notes:            -O            optimizer
notes:            -g            debugger
notes: some versions of the HP C compiler optimizer breaks snefru.c!
notes:       consider recompiling this file seperately without optimization
notes:       (-spaf and genek)
notes:       *** Some people are having considerable difficulty getting
notes:       *** getting Tripwire to run using gcc.  Please see the
notes:       *** ./contrib/README.hpux file for details.
notes:       support for CDF added by Cory Cohen.  see ./contrib/README.cdf
```

Aquí encontramos información como : versiones de S.O. que son soportadas, el compilador, banderas del compilador y el archivo que se debe usar para la configuración, después de tomar esta información, editar el archivo Makefile, y poner los valores adecuados en las siguientes variables :

LEX, YACC, CC, CFLAGS, y los siguientes:

DESTDIR, que es el directorio donde estarán los binarios y,
MANDIR, que es el directorio donde se almacenan los manuales.

2) El siguiente archivo a modificar es el config.h, el cual se localiza en el directorio ./include. La línea a modificar es donde se especifica el archivo de configuración de nuestro sistema operativo, por ejemplo, si en nuestro caso es hp-ux, entonces quedaría de la siguiente manera :

```
#include "../configs/conf-hpux.h"
```

3) Por último se copia el correspondiente archivo tw.config (de configuración) al directorio donde apunta la variable CONFIG_PATH, que se encuentra definida en el archivo config.h, debajo del directorio ./include. Para este ejemplo se copiará el archivo tw.conf.hpux.

```
# cp . /tripwire-1.2/configs/ tw.conf.hpux  $CONFIG_PATH/tw.config
```

En este archivo se definen los directorios que se quieren verificar, tiene un contenido similar a éste :

```
# cd $CONFIG_PATH
# vi tw.config
```

```
# $Id: tw.conf.hpux,v 1.3 1993/08/19 05:27:12 genek Exp $
# Lance R. Bailey <lrb@ctrg.rri.uwo.ca>
# First, root's "home"
/.rhosts      R    # may not exist
/.profile     R    # may not exist
/.cshrc       R    # may not exist
/.login       R    # may not exist
/.forward     R    # may not exist
/stand        R

# Now, some critical directories and files
# Some exceptions are noted further down
/etc/inetd.conf R
/etc/rc         R
/etc/gettydefs  R
/etc/exports    R
/etc/motd       L
/etc/rmtab      L
/etc/utmp       L
/etc/group      R    # changes should be infrequent
/etc/passwd     L

# Checksumming the following is not so critical. However,
# setuid/setgid files are special-cased further down.
/bin         R-2
/usr/bin     R-2
/usr/lib     R-2
=/usr        L
=/usr/spool  L
/usr/spool/cron      L
/usr/spool/mqueue    L
/usr/mail            L

# Here are entries for setuid/setgid files. On these, we use
# both signatures just to be sure.

/bin/df        R
/bin/ipcs      R
/bin/login     R
/bin/mail      R
/bin/passwd    R
/bin/rmail     R
```

```
/bin/su         R
/bin/write      R
/var           R-12
/oracle         R-12
/home           R-12
```

Las letras L y R son plantillas que indica que a esos directorios se les hará una revisión. En esta tesis se usará la R la cual sirve para verificar los cambios en permisos, en el número de inodo, el número de ligas, uid, gid, timestamp de creación y modificación, y con el 12 se indica que no utilice firmas de autenticación al momento de la verificación de los archivos o directorios.

Después de haber armado la configuración, se procede a la compilación del software con el comando make :

```
# make
```

Ahora que ya se generaron los binarios, se procede a construir la base de datos de la siguiente manera :

```
# ./tripwire  -initialize
```

```
### Phase 1:  Reading configuration file
### Phase 2:  Generating file list
./tripwire: /.rhosts: No such file or directory
./tripwire: /.cshrc: No such file or directory
./tripwire: /etc/motd: No such file or directory
./tripwire: /usr/etc/rpc.rwalld: No such file or directory
### Phase 3:  Creating file information database
./tripwire: /var/spool/sockets/pwgr/client5590: disappeared.  Skipping...
./tripwire: /var/spool/sockets/pwgr/client5589: disappeared.  Skipping...
```

Después de generar la Base de Datos, las siguientes revisiones se ejecutan de la siguiente manera :

```
# ./tripwire
```

```
### Phase 1:  Reading configuration file
### Phase 2:  Generating file list
./tripwire: /.rhosts: No such file or directory
./tripwire: /.cshrc: No such file or directory
./tripwire: /.login: No such file or directory
./tripwire: /.forward: No such file or directory
./tripwire: /usr/etc/rpc.rwalld: No such file or directory
### Phase 3:  Creating file information database
./tripwire: /var/spool/sockets/pwgr/client5606: disappeared.  Skipping...
./tripwire: /var/spool/sockets/pwgr/client5607: disappeared.  Skipping...
### Phase 4:  Searching for inconsistencies
###               Total files scanned:        31799
###                   Files added:         2
###                   Files deleted:       2
###                   Files changed:        11618
###               After applying rules:
###                   Changes discarded:       11612
###                   Changes remaining:       14
added:  -rwxrwxrwx 0           0 Apr  9 19:38:57 2002 /var/spool/sockets/pwgr/client5608
deleted: -rwxrwxrwx 0           0 Apr  9 19:30:57 2002 /var/spool/sockets/pwgr/client5602
changed: -rw-r--r-- 0      332314 Mar 12 19:58:08 2002 /etc/opt/resmon/log/client.log
changed: drwxrwxrwx 2        1024 Apr  9 19:41:56 2002 /tmp
changed: drwxrwxrwx 0        3072 Apr  9 19:42:30 2002 /var/spool/sockets/pwgr
changed: -rw-r----- 400    843776 Apr  9 19:42:36 2002 /oradatos/ctls/control01.ctl
changed: drwx------ 101      1024 Mar 19 20:17:49 2002 /home/fabian
### Phase 5:  Generating observed/expected pairs for changed files
/etc/opt/resmon/log/client.log
    st_ctime: Tue Apr  9 19:36:57 2002     Tue Apr  9 19:04:56 2002
/var/rbootd/C000000.reply
    st_mtime: Tue Apr  9 19:38:49 2002     Tue Apr  9 19:31:17 2002
    st_ctime: Tue Apr  9 19:38:49 2002     Tue Apr  9 19:31:17 2002
```

```
/var/spool/pwgr/status
    st_mtime: Tue Apr  9 19:41:54 2002    Tue Apr  9 19:30:57 2002
    st_ctime: Tue Apr  9 19:41:54 2002    Tue Apr  9 19:30:57 2002
/var/spool/sockets/pwgr
    st_mtime: Tue Apr  9 19:42:30 2002    Tue Apr  9 19:31:28 2002
    st_ctime: Tue Apr  9 19:42:30 2002    Tue Apr  9 19:31:28 2002
/oradatos/ctls/control01.ctl
    st_mtime: Tue Apr  9 19:42:36 2002    Tue Apr  9 19:31:40 2002
    st_ctime: Tue Apr  9 19:42:36 2002    Tue Apr  9 19:31:40 2002
/oradatos/ctls/control02.ctl
    st_mtime: Tue Apr  9 19:42:36 2002    Tue Apr  9 19:31:40 2002
    st_ctime: Tue Apr  9 19:42:36 2002    Tue Apr  9 19:31:40 2002
/oracle/product/network/log/listener.log
    st_mtime: Tue Apr  9 19:41:52 2002    Tue Apr  9 19:21:51 2002
    st_ctime: Tue Apr  9 19:41:52 2002    Tue Apr  9 19:21:51 2002
/export/home/fabian
    st_mode: 40700                  40755
    st_ctime: Tue Apr  9 19:40:29 2002    Tue Mar 19 20:17:49 2002
```

A partir de esta salida se analiza que archivos o directorios han sido modificados, ya que viene de manera detallada toda esta información.

3.6.3 Saint

En el capítulo anterior se mencionó que es una herramienta de monitoreo a posibles huecos de seguridad en los servidores, requiere de ciertos requisitos para ser usada :

1: Tener instalado perl 5.0 o posterior
2: Para poder correr saint se necesita un navegador (netscape, mosaic, cimera, etc)
3: Un servidor de gran capacidad por la revisión que realizaría a varios servidores o a una red completa de servidores

Cubriendo estos requisitos se procede a la instalación :

```
# gunzip saint-3.1.tar.gz
# tar xvf saint-3.1.tar
```

Antes de realizar la compilación se debe correr un script que genere el archivo Makefile para el sistema operativo en donde se esté instalando :

```
# ./configure
```

Después de haber realizado el archivo de configuración, se procede a la compilación del software con el comando make :

```
# make
```

Después se instala la página de ayuda con :

```
# make install
```

Una vez instalado el producto se corre el binario generado, el cual se localiza en la misma ruta donde se descompactó el archivo tar :

```
# ./saint
```

Este llamará de manera automática al navegador para que pueda mostrarse el saint, cuya página principal se muestra en la figura 3.1 :

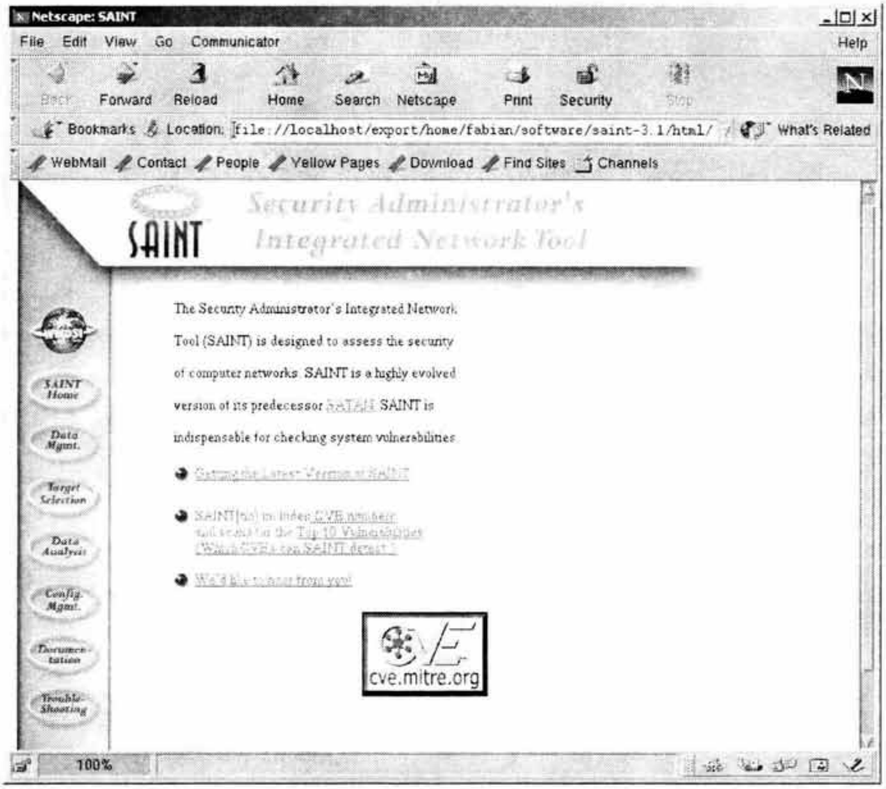


Figura 3.1

En el menú del lado izquierdo se selecciona la opción de target selection y presenta la siguiente página (figura 3.2) :

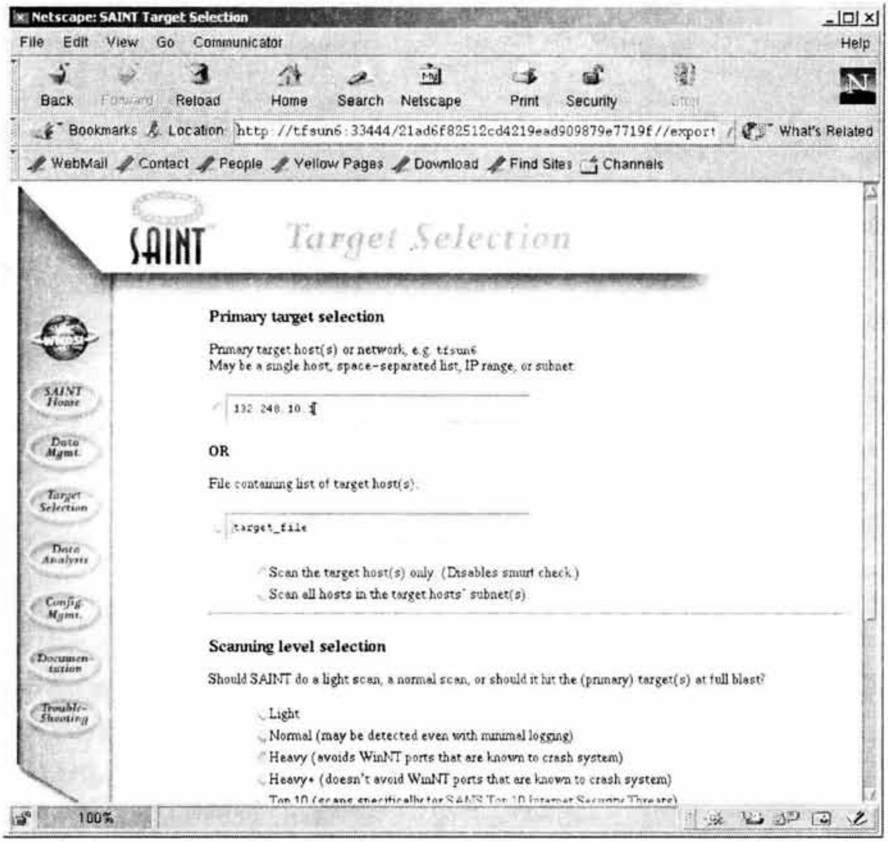


Figura 3.2

Aquí se indica la dirección IP del servidor al cual se le hará la revisión, en este caso se seleccionó el 132.248.10.1, más abajo viene el tipo de revisión que se le aplicará : light, normal, heavy, heavy +, etc, cada una tiene un tipo de verificación más específico, con esto, se puede empezar la revisión (con el botón de start scan), a continuación manda los resultados, como se muestra en la figura 3.3 :

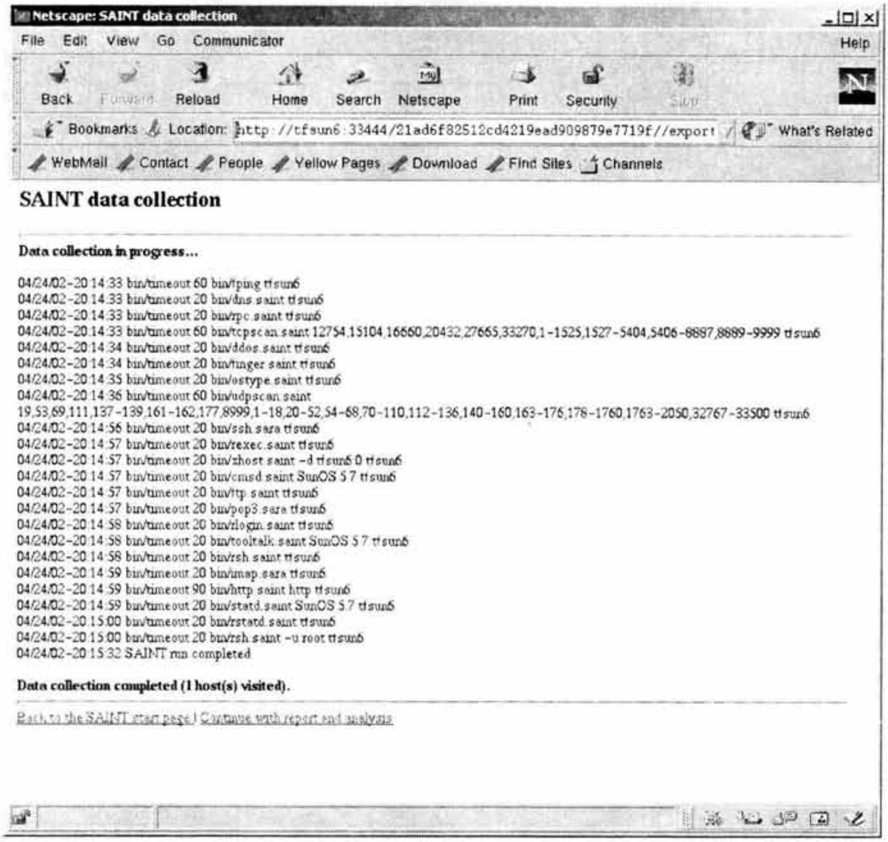


Figura 3.3

Aquí se pueden ver algunos servicios que tiene habilitado el servidor que se probó en este ejemplo. Esta herramienta es muy poderosa y de muy fácil uso, se puede analizar cada uno de los huecos de seguridad que encontró e inclusive propone algún tipo de solución o mejora en los detalles que encuentre. También clasifica los posibles huecos de seguridad y las consecuencias que estos puedan originar.

3.7 OTRAS HERRAMIENTAS

3.7.1 sudo

La instalación de sudo consiste de lo siguiente :

```
# gunzip sudo-1.6.3p6.tar.gz
# tar -xvf sudo-1.6.3p6.tar
```

Correr el script configure, el cual es un shell que determinará el ambiente del sistema para poder construir sudo.

```
# ./configure
```

Después de haber armado la configuración, se procede a la compilación del software con el comando make :

```
# make
```

El siguiente paso es la instalación de sudo y su documentación :

```
# make install
```

Con todo lo anterior ya se tiene instalado el producto sudo, si no se reportaron errores durante la instalación, se verifica que existan los siguientes archivos :

/usr/local/bin/sudo: Es el comando con el cual se le permitirá a un usuario ejecutar comandos de superusuario
/usr/local/sbin/visudo: Con este comando se podrá editar el archivo /etc/sudores, similar al comando vi
/etc/sudoers: Es un archivo de configuración que contiene la lista de usuarios que podrán ejecutar ciertos comandos aquí descritos.

Se muestra un ejemplo de cómo utilizar esta herramienta:

Primero se tiene que editar el archivo /etc/sudoers para otorgar los permisos a los usuarios deseados:

```
# visudo  /etc/sudoers
```

Una vez abierto este archivo se le añaden los usuarios y los comandos que serán permitidos, similar al siguiente:

```
# sudoers file.
```

```
user ALL=/usr/sbin/useradd, /usr/bin/date, /usr/bin/ocio.sh
```

En este ejemplo en particular se especifica que podrá usar los comandos useradd, date y el archivo ocio.sh. El primer comando solo root lo puede ejecutar, el segundo lo puede usar cualquier usuario, pero no para modificar la hora del sistema y el archivo ocio.sh es un script con permisos para que solo root lo pueda ejecutar. Con sudo, el o los usuarios especificados podrán ejecutar este tipo de comandos sin que se les otorguen privilegios de root.

Una vez que ya se configuró el archivo anterior, el usuario asignado podrá hacer uso de sudo de la siguiente manera : después de haber establecido una conexión vía telnet o secure shell, teclea :

```
$ sudo useradd
Password:
```

En este momento pide teclear la contraseña del usuario, no la de root, ahora si se podrá usar:

```
$ sudo useradd .........
```

Es necesario poner sudo antes del comando deseado, ya que si no, no funcionará.

3.7.2 Nullshell

Esta herramienta tiene la característica de que se instala de manera sencilla y la configuración se realiza rápidamente, sus características se vieron en el capítulo anterior.
La instalación de nulshell consiste de lo siguiente :

```
# gunzip nullshell.tar.gz
# tar xvf nullshell.tar
```

Después de que se descompacta el archivo tar se genera el directorio ./nullshell, el cual solo contiene 2 archivos, uno de ayuda y el fuente del programa (README y nullshell.c).

Antes de realizar la compilación se deben hacer los siguientes cambios al archivo nullshell.c :

línea 37 : Aquí se encuentra la ruta de la bitácora, en ésta se registrarán todos los intentos de acceso, se puede dejar el que viene por default o se puede cambiar este valor.

```
#define FOAD_LOG "/usr/adm/foad.log"
```

línea 95 : Éste es el mensaje que se mostrará cuando se quiera acceder a una cuenta deshabilitada, se puede dejar la que viene por default o cambiar este mensaje:

```
printf( "\n\nCUENTA INHABILITADA.\n\n");
```

Después de realizar estos cambios, se procede a la compilación de la herramienta:

```
# cc -o nullshell nullshell.c
```

Esto generará el archivo binario llamado nullshell, el cual deberá ser movido de preferencia a donde están los demás shells y cambiarle los permisos:

```
# mv ./nullshell /bin/nullshell
#chmod 555 /bin/nullshell
```

También cambiarle los permisos a la bitácora:

```
# chmod 666 /usr/adm/foad.log
```

En algunos sistemas operativos es necesario escribir la ruta de este shell, en el archivo /etc/shells. Aquí termina toda la instalación, para usar la herramienta hacer lo siguiente:

En el archivo /etc/passwd, cambiarle el shell a la cuenta que se desea deshabilitar:

 # vi /etc/passwd

```
fabian:x:104:10:Rosendo Fabian Hernandez:/export/home/fabian:/bin/nullshell
```

Realizar una conexión:

 # telnet anita

```
login: fabian
Password:
Last login: Thu Apr 11 19:59:57 from 10.150.37.180
CUENTA INHABILITADA
```

Por último se revisa la bitácora para verificar que se haya registrado este intento de conexión:

 # more /usr/adm/foad.log

```
02-04-11:20:27:10 login by fabian (Rosendo Fabian Hernandez) from pts/2
```

Con esto se puede saber si algún usuario que ha sido deshabilitado, sigue haciendo intentos de conexión.

4. CAPITULO 4: PROBLEMÁTICA ACTUAL DE LOS SERVIDORES DE UNA ENTIDAD GUBERNAMENTAL.

4.1 INTRODUCCIÓN

En los capítulos anteriores se definieron puntos esenciales como administración del Sistema operativo Unix, su seguridad básica pero elemental, características de herramientas de libre distribución para la seguridad de Unix, la instalación y configuración de estas herramientas, todo esto para llegar al objetivo principal de esta tesis, que es resolver la actual problemática de esta entidad gubernamental a nivel de seguridad en los servidores Unix.

En este capítulo se definirá todo lo que rodea y afecta o puede afectar a la problemática actual, como : factores internos, factores externos, crecimiento del problema, causa de la investigación, consecuencias que puede tener la entidad, efectos que puede arrojar el problema y otros puntos que se abordarán en su momento.

4.2 MARCO TEÓRICO

Los administradores de servidores Unix, ya sea de pocos servidores o de grandes centros de cómputo, dejan a un lado el problema de la seguridad, debido al exceso de confianza o a la falta de información referente a este tema, lo cual en la actualidad es un gran error, por las consecuencias tan graves que esto podría ocasionar.

Cuando se menciona acerca de la seguridad en los servidores, se refiere a que una buena administración no es solamente tener funcionando las aplicaciones, si no, hacer que el servidor funcione tal como lo espera el usuario, esto no es una tarea tan compleja, sin embargo, al hablar sobre seguridad, ésta se complica, ya que se requiere de un análisis de funcionamiento del servidor y la instalación de software de herramientas de seguridad, que pudieran afectar el "performance" de éste, pero casi sin que sea percibido. Es muy frecuente que al implantar seguridad los usuarios se vean afectados en cuanto a políticas que se les imponen, pero es necesario inculcarles un poco de esta cultura, si lo que queremos es mantener la información protegida de estos mismos usuarios.

Con estas características debe contar una implantación de seguridad (figura 4.1) :

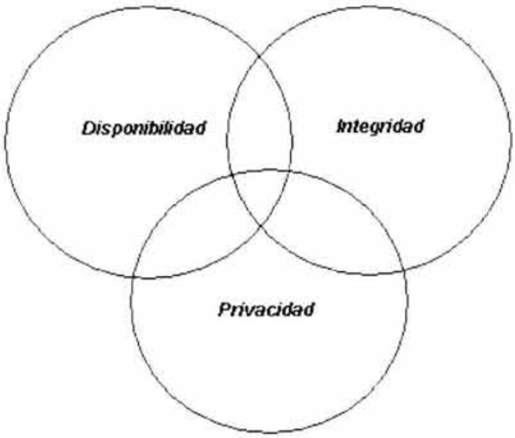


Figura 4.1

Disponibilidad : Éste punto se refiere a que la información del servidor siempre debe estar disponible, además de ser confiable, esto es, que no existan datos o información que hayan sido alterados. Algunos puntos de falla son : la ausencia de respaldos o que alguna persona sin intención borre información importante.

Integridad : Éste es un punto demasiado importante, ya que se refiere a mantener la integridad de la información, esto es, no deberá ser alterada en su contenido o sus permisos. Algunos puntos de falla son : problemas de algún software (bugs) o que alguna persona malintencionada modifique información importante.

Privacidad : Es muy importante mantener la privacidad o confidencialidad, teniendo una apropiada autenticación, autorización y permisos.

Cada uno de los 3 puntos anteriores son susceptibles. Si los permisos no son correctos o los usuarios no usan apropiadamente sus contraseñas, la confidencialidad de la información puede ser comprometida. Si el software no es debidamente parchado o no se verifican los cambios que sufren los archivos importantes para el sistema, entonces la integridad de los datos puede ser afectada. Si las medidas de protección en fallas de hardware o desastres físicos no son tomadas en cuenta, la disponibilidad de la información está en riesgo.

Para comenzar a diseñar uno o mas esquemas de seguridad, es necesario comprender el problema principal que afecta a los servidores que se desean proteger, para el caso de esta tesis se detectó la necesidad de cuidar los equipos del entorno que le rodea, por la gran importancia de la información que se maneja, además de dar a conocer el uso de algunas herramientas de seguridad de libre distribución y los grandes beneficios que estas brindan sin costo alguno, todo esto también sirve como apoyo para tener un panorama más amplio de la seguridad en cómputo.

Fue necesario la investigación de las herramientas de seguridad (de libre distribución), su clasificación y su propósito, para poder estructurar bien el tema principal de esta tesis.

En este capítulo se recopilarán todos los datos relevantes que sean de utilidad y que apoyen al diseño de esquemas que se verán en el ultimo capítulo.

4.3 ANÁLISIS E IDENTIFICACIÓN DE LA PROBLEMÁTICA ACTUAL DE LA ENTIDAD GUBERNAMENTAL

La seguridad es una alternativa para cualquier administrador que desee proteger su entorno computacional, pero no sólo es decidir hacerlo, si no, se requiere de un análisis del lugar donde labora, determinar si se puede modificar un poco el entorno, si es así, evaluar si no se ven afectadas las aplicaciones de los servidores, pero lo más importante es el esfuerzo del administrador por lograr su objetivo, que en este caso será la protección de su entorno. Para esta tesis se realizó un análisis del entorno y sus características.

El entorno original de esta entidad gubernamental, se muestra en la figura 4.2.

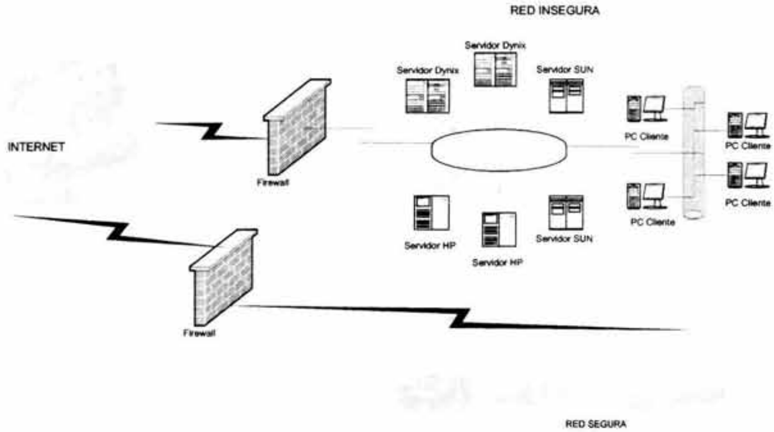


Figura 4.2

Algunas características del entorno desprotegido (red insegura) :

- Servidores Unix de diferentes plataformas
- Servidores NT y computadoras personales con sistema operativo Windows X.
- Todos estos equipos se encuentran dentro de un mismo entorno,
- No existen firewalls o algún dispositivo que impida la comunicación directa entre ellos.
- Algunos servidores Unix están en producción
- Todos los servidores son de muy poca capacidad.
- Soportan la ejecución y transferencia de grandes cantidades de información.
- Aislamiento de la red segura
- Cuenta con servicios que son obvios puntos de ataque (http, telnet, smtp, ftp, nfs, etc).
- Firewall (a nivel de software) entre el entorno inseguro e internet.
- Ningún tipo de herramienta instalada para conexión segura.
- Acceso a bases de datos de manera directa a través de clientes gráficos.

De principio se pensó en cambiar todo lo que esta dentro del entorno inseguro, sin embargo, después de la adquisición de los nuevos equipos se optó por conservar los otros servidores, pero en otro entorno.

Algunas características del entorno protegido (red segura) :

- Equipos de cómputo de mucho mayor capacidad.
- Servidores Unix homogéneos.
- Infraestructura moderna tanto en hardware como en software.
- En lo que se refiere a software (en la parte de seguridad), se implantó una suite completa.
- Todos los servicios están protegidos con la suite de seguridad.
- Cuenta con un monitoreo constante que verifica fallas o posibles fallas.
- Protección con firewalls a nivel de hardware.

En la implementación del nuevo entorno de red, la seguridad fue una de las piezas claves y factor esencial en el diseño de éste, para la protección de la información, entonces fue necesario una investigación de la entidad gubernamental, determinando la importancia y relevancia de la información que se iba a transferir, pero se descartó totalmente el otro entorno, debido a que no se tenia contemplado que siguiera funcionando.

Para saber si la seguridad es indispensable se deben tomar en cuenta los antecedentes mencionados en el capítulo 1, la importancia de la información almacenada cotidianamente, lo vulnerable que quedará el entorno desprotegido, el personal que labora en la empresa, las limitaciones de los equipos actuales, etc, por lo tanto, es indispensable tomar medidas preventivas antes que suceda algún percance o ataque dentro de esta dependencia.

Lo mencionado en el párrafo anterior se debe al crecimiento de usuarios en los servidores actuales, sus conocimientos (en algunos usuarios limitado y en otros de un nivel muy avanzado), el volumen de la información y los servidores con sistemas operativos muy atrasados.

A continuación se muestran y detallan puntos vulnerables que les pueden afectar a servidores de cómputo :

- ***Usuarios***
- ***Respaldos***
- ***Tipo de administración***
- ***Archivos de configuración***
- ***Instalación de software de aplicación***
- ***Instalación de parches***
- ***Ingeniería social***
- ***Limitaciones de los servidores***
- ***Ubicación de los entornos***
- ***Vulnerabilidad en el entorno desprotegido***

Los usuarios se consideran como las principales causas de vulnerabilidad dentro de los sistemas de cómputo [1] :

4.3.1 Usuarios de producción

Uno de los principales factores en la entidad, es la cantidad de personal que labora en ella y las funciones que realiza cada área, tomando en cuenta que tienen acceso al sistema operativo Unix y a las Bases de datos que están en los servidores. Muchos de estos usuarios acceden a programas o a sistemas por medio de cuentas Unix, esto es, primero establecen una sesión de Unix y luego se conectan a la Base de datos (para esta tesis son aproximadamente 1000 concurrentes) [2]. El riesgo que se tiene con estas cuentas es que se conectan al sistema operativo a través de programas como telnet o algunos otros programas que actúan como terminales, después acceden a la Base de datos que les corresponde.

Esto es demasiado riesgoso si se toma en cuenta que la mayoría de los usuarios son personas que no conocen comandos del sistema operativo Unix y que no saben el riesgo que implica prestar sus cuentas o dejar sus sesiones abiertas, que es lo que sucede muy a menudo en cualquier dependencia o empresa y que no se puede controlar, a pesar de capacitarlos, de implantarle políticas, aplicarles sanciones, etc., todo esto es inevitable y es una de las cosas más importantes que se deben de considerar en implantaciones posteriores, por que se puede proteger el sistema a un buen nivel de seguridad, pero si no se considera que el usuario es uno de los puntos más vulnerables que existe en los servidores, no se podrán proteger estos eficientemente.

4.3.2 Usuarios de respaldos (Ejecución de scripts)

En todo lugar donde se lleva una buena administración se le da una alta prioridad a la ejecución de respaldos, sin embargo, esto no indica que se realicen de manera adecuada o sin tener cuidado en los privilegios otorgados a los usuarios que elaboran estos o a los programas (scripts) que ejecutan; estos privilegios se otorgan debido a que regularmente un administrador no elabora respaldos ya que estos se llevan a cabo "normalmente" después de la producción, esto es, en horas no laborables, que en la mayoría de los casos es por la noche.

Uno de los peligros al ejecutar los respaldos, es que estos se tienen que efectuar con privilegios de administrador para poder respaldar toda la información (filesystems completos, bases de datos o archivos específicos), entonces la manera más fácil de librarse de este problema es la creación de un usuario alterno pero con UID = 0, pero esto indica que a pesar de ser un usuario alterno, al tener este UID, automáticamente tiene todos los privilegios del administrador, y este no es el único problema, si no que normalmente elaboran respaldos de manera remota, esto es, por conexiones a través de telnet. Estos usuarios generalmente solo actúan como operadores del Sistema Operativo Unix, sin tener conocimientos más avanzados, entonces se está corriendo el gran riesgo de ejecutar un comando que pueda afectar al sistema, aunque lo haga sin intención alguna, tal vez en ese momento no se de cuenta del daño que le está causando al sistema que se supone solo está respaldando. Otra opción que generalmente se usa para realizar respaldos es por medio de elaboración de scripts,

1 La mayoría de los usuarios finales carecen del conocimiento necesario para manejar equipos de cómputo, sobretodo Unix (sistema operativo requerido para que la mayoría de las aplicaciones de la actualidad), por eso se consideran el principal punto de falla.
2 Las estimaciones del número de usuarios finales fueron tomadas de los cálculos hechos por una o varias personas entrevistadas. El propósito de incluir estas cifras es dar una idea de la concurrencia, esto es que ejecuten procesos al mismo tiempo.

los cuales adquieren privilegios para poder respaldar el sistema completo. Estos scripts se realizan sin cuidado alguno y confiando plenamente en las personas que lo ejecutan, sin percatarse de que quien acceda a esta cuenta (la de respaldo) podrá ejecutar este script y adquirir privilegios de administrador.

Otro punto importante es la revisión de los respaldos, debido a que normalmente se elaboran y se procede a guardar las cintas en alguna bodega o algo por el estilo, sin antes haber verificado que el respaldo se haya efectuado de manera correcta o si la cinta se encuentra en buen estado

También es muy común encontrar sin etiquetas a las cintas, olvidándose que es necesario tener etiquetadas las cintas, para que en una emergencia se obtenga de manera inmediata la cinta, el respaldo mas reciente y recuperar en el menor tiempo posible lo deseado. Otro aspecto a considerar es que se debe conocer las propiedades de la cinta, como : el tiempo de vida de una cinta, espacio máximo a ocupar en la cinta y tipo de cinta.

En algunos servidores tal vez no se le pueda dar solución por medio de herramientas de seguridad, ya que, por falta de capacidad, por tener una versión de sistema operativo demasiado atrasada o falta de librerías, no se opta por asegurar el servidor con solo permisos o restricción de uso de comandos, los cuales se van a proponer en uno de los esquemas que se mencionarán en el último capítulo.

4.3.3 Usuarios administradores (Administración descentralizada)

El punto de partida para tener seguridad en los servidores Unix es la buena administración de estos, pero cuando no se esta administrando de manera adecuada ya sea por la inexperiencia del administrador o por la falta de conocimientos de este, entonces hay un grave problema, mas aun cuando se tiene mas de un administrador el problema se vuelve mas grave. Lo anterior se refiere a que es muy común que para facilitar la administración en los servidores se opta por tener una administración descentralizada, esto es, que mas de una persona lleve la administración de un servidor y esto implica que se pierda el control de los cambios que se realicen.

En todas las empresas existe mas de un administrador, por diferentes razones : ya sea por evitar que solo una persona cuente con estos privilegios y no depender solo de esta persona, para evitar una administración totalmente centralizada, por que los mandos superiores desean tener los privilegios necesarios a pesar de no contar con los conocimientos indispensables, por mencionar algunas. Esto solo conduce a que se pierda la confidencialidad y el control, ya que cuando sucede algún percance es imposible saber quien fue el que ocasiono el desperfecto o peor aún si ni siquiera fue alguno de los tantos que tienen la cuenta de administrador y fue alguien que obtuvo esta cuenta por algún medio.

Esto se considera muy riesgoso, por todas las modificaciones que se pueden hacer al servidor, respecto a : archivos de configuración del sistema, instalación de software de aplicación comercial o propio, manejo de bitácoras, alta y baja de usuarios, daño al servidor, cambio de parámetros (del kernel), permisos a archivos, permisos de acceso, habilitación de servicios de red y, por lo tanto, que se pierda todo el control de la administración.

Es recomendable cuando se tiene mas de un servidor, distribuir la administración entre 2 o mas administradores, cada uno en un servidor, o si es necesario, que cada administrador se encargue de cosas especificas dentro de un mismo servidor y así se tendrá un mejor control. Esto se puede lograr por medio de programas que restringen el uso de comandos de administración.

4.3.4 Archivos de configuración del sistema

En el sistema existen muchos archivos de configuración del sistema para : la red, compartir recursos, la seguridad del sistema, el manejo de bitácoras, el tipo de montaje de los filesystems, los servicios que levanta en cada nivel, entre otros, todos estos archivos son relevantes para el sistema, ya que de estos toma la configuración al momento de estar levantando o al tirar y levantar los servicios y/o demonios que ocupan estos archivos. Entonces cuando se modifica alguno de estos archivos (necesariamente como administrador) y no se hace un previo respaldo, se corren riesgos como perder la configuración original y no poder rescatarla, borrar un archivo importante, los cambios realizados pueden afectar el rendimiento del servidor, incluso modificar un archivo casi al mismo tiempo e inclusive dar de baja el servidor sin avisar a los demás administradores.

4.3.5 Instalación de software de aplicación comercial o propio

La instalación de software o paquetes es un punto relevante por que puede afectar el desempeño del servidor y otras aplicaciones previamente instaladas. También se tiene el problema de que con las aplicaciones propias, en ocasiones dependa de algún software comercial, regularmente compiladores, y como se vio en capítulos anteriores, esto es un gran hueco de seguridad, sin embargo, es muy común que siempre existan áreas que desarrollen programas.

Otro punto importante y que debe saber un administrador, es que antes de empezar a instalar algún software, se tiene que asegurar de que la versión del sistema operativo que se tiene instalado en el sistema sea compatible con el software de aplicación que se va a instalar, o si es necesario instalar un parche o algún otro software adicional, para evitar que haya conflictos con el sistema. También tiene que investigar los requerimientos mínimos necesarios en cuanto a hardware y software que necesite la aplicación a instalar, además de que si es necesario modificar parámetros del kernel, si es así, entonces verificar que sus valores sean correctos y soportados por el servidor, para prevenir probables fallas y contemplar que el servidor se tenga que dar de baja, a menos que el kernel sea dinámico y no sea necesario. Una recomendación importante es que antes de empezar a instalar un nuevo software, se respalde completamente el servidor, por si llegase a fallar algo, se deje el servidor como estaba antes de la instalación del nuevo software.

4.3.6 Instalación de parches

Este punto va muy relacionado con el anterior, pero éste es más sobresaliente ya que la instalación de los parches si le pueden afectar al sistema, si no se le instalan los adecuados y con una previa prueba en un servidor que no sea crítico. Algunos parches modifican valores del kernel y por lo tanto es necesario que el servidor se reinicie (en algunas versiones actuales de sistemas operativos el kernel ya es dinámico, esto es, no requiere que se reinicie el servidor) para su recompilación de éste, y suele suceder que administradores novatos no se percaten de esto y aquí se produzca una situación grave, como el reiniciar el servidor en plena producción.

4.3.7 Ingeniería Social

Se denomina ingeniería social a las técnicas usadas a través del engaño sobre personas para obtener información de las debilidades de un sistema. La ingeniería social puede ser el eslabón más débil de cualquier cadena de políticas de seguridad. Se dice que la única computadora segura es aquella que nunca será encendida. Intentar que alguien suministre su número de tarjeta de crédito, suena poco factible, sin embargo puede brindar datos confidenciales diariamente en diferentes medios, como el papel que arroja a la basura o el papel adhesivo (post-script) con su contraseña debajo del teclado o enfrente de su monitor. El factor humano es una parte esencial en la seguridad, ya que no existe un sistema que no dependa de algún dato ingresado por un operador humano. Esto significa que esta debilidad de seguridad es universal, independientemente de la plataforma, el software, la red o el equipo. Cualquier persona con el acceso a alguna parte del sistema, es un riesgo potencial de inseguridad y también puede tener las herramientas para intentar una ingeniería social, la única diferencia es la habilidad y conocimientos al hacer el uso de estas herramientas.

Existen métodos para que una persona pueda lograr su objetivo :

Un método es simplemente con preguntas directas, donde a un individuo se le pide información pero probablemente no tendrá éxito, sin embargo es el método más fácil. Otro método es ejecutado indirectamente en una situación previamente preparada, esto involucra mucho más trabajo para la persona que hace el esfuerzo del cuestionamiento, pero obtiene un conocimiento extenso del objetivo. Una de las herramientas esenciales usadas para la ingeniería social es una buena recolección de los hábitos de los individuos.

La ingeniería social se dirige a las personas con menos conocimientos, dado que los argumentos y otros factores tienen que ser construidos generando una situación creíble para que la persona la ejecute.

Algunos ejemplos con los que se puede obtener información :

- La ejecución de un virus troyano por parte del usuario, adjunto a un correo electrónico enviado con un subject que sea familiar o simplemente con un interesante título al destinatario como "es divertido, pruébalo", etc.
- La voz agradable de un hombre o mujer, que pertenece al soporte técnico de nuestra empresa o de nuestro proveedor de tecnología, que no requiera telefónicamente de información para resolver un inconveniente detectado en nuestra red.
- El llamado de un usuario que necesita que se le asignen nuevamente su clave porque la ha cambiado durante el transcurso del día y no la recuerda.

Esto son ejemplos, citados simplemente para mostrarle a : los administradores del sistema, analistas de seguridad, técnicos, las personas a las que se le confían herramientas de trabajo esenciales o comunicación, que están muy envueltas en los ataques diseñados por expertos de las computadoras.

Entonces se deben de tomar medidas como crear conciencia de la seguridad a las personas que formen parte del trabajo (aunque no tengan acceso a la computadora), la mejor defensa contra esto, es la educación y se logra explicando a los empleados la importancia de la seguridad de la computadora y sus datos, advirtiéndoles que son responsables directos de su contraseña y de lo que hagan con ella.

Como se vio en los párrafos anteriores es más fácil utilizar a las personas, que explotar vulnerabilidades o malas implementaciones de un sistema y que requiere esfuerzo educar a los usuarios para que puedan prevenirse de la ingeniería social.

4.4 ¿POR QUÉ ES UN PROBLEMA?

En el tema anterior se identificaron los puntos vulnerables que puede tener un servidor y que tanto se puede afectar a la entidad con estos, también se dieron algunas recomendaciones para evitar esas vulnerabilidades, ahora se identificará por que se puede catalogar como un problema, en donde esos puntos afectan directamente a la entidad, además de otros puntos de falla [3].

4.4.1 Limitaciones de los servidores

Uno de los grandes problemas es el tipo de servidores con los que cuenta esta entidad, estos son de diferentes plataformas y de capacidad muy limitada, lo que origina problemas con el software instalado y la compatibilidad entre ellos. En algunos casos se tiene que instalar software idéntico, pero de versión diferente o en ocasiones instalar una base de datos de versión muy atrasada para que el formato de los datos entre las bases puedan ser compatibles. Todo esto se debe a la imparcialidad de hardware y software entre los servidores, debido a que se adquirieron con algunos años de diferencia, lo que ocasiona que no haya mucha compatibilidad entre ellos.

Estos servidores son de diferentes plataformas, de poco espacio en disco, memoria muy limitada (apenas para poder correr las aplicaciones instaladas), CPU de baja velocidad e interfaces de red que transmiten a baja velocidad. A continuación se muestra una tabla en donde se puede identificar la limitación de los servidores :

Servidores	*Espacio promedio en disco*	*Memoria*	*Velocidad CPU*	*Tarjeta de red*
Sun	**4 GB**	**64**	**360 Mhz**	**100 Mbps**
IBM	**1 GB**	**64**	**60 Mhz**	**100 Mbps**
HP	**4 GB**	**256**	**400 Mhz**	**100 Mbps**

En la tabla anterior se puede notar que el acceso a los servidores es por tarjetas de red a 100 Mbps, que no es nada despreciable, sin embargo, si ocasiona cuellos de botella en algunos servidores. También se muestra que la velocidad del procesador es muy baja, lo que origina menor tiempo de respuesta en todos los procesos del servidor. Otro grave problema es la memoria, sobretodo por que las bases de datos consumen gran parte de esta y dejan un porcentaje muy bajo a las demás aplicaciones y como casi no hay espacio en disco, no se puede alojar mas memoria swap, además, de que también el software de base de datos ocupa casi 500 MB en disco, lo que significa que en los servidores con un disco de 1GB, solo deja la mitad disponible. Un último problema radica en los accesos que se tienen a disco y el tiempo de vida de estos, que hace que crezca la probabilidad que fallen.

4.4.2 Ubicación de los entornos

En la etapa de análisis los entornos son muy importantes, por que estos son los que originan todo lo que rodea esta tesis. Lo anterior se menciona por la creación de 2 entornos independientes (figura 4.3), uno de ellos cuenta con modernas instalaciones y está protegido con software de seguridad comercial y el otro entorno tiene instalaciones un poco obsoletas, totalmente desprotegido y aislado del primer entorno.

3 Los datos utilizados corresponden a los años de 2002 y 2003.

La decisión de aislamiento total estuvo basada en que, en primera los servidores que estaban en producción todavía son de utilidad, en segunda, no necesitan estar comunicados con los nuevos servidores y el último punto y más importante se refiere al tema de la seguridad, por que mientras los nuevos servidores están totalmente protegidos contra cualquier ataque, los servidores antiguos se catalogan como muy vulnerables y de uso no tan crítico.

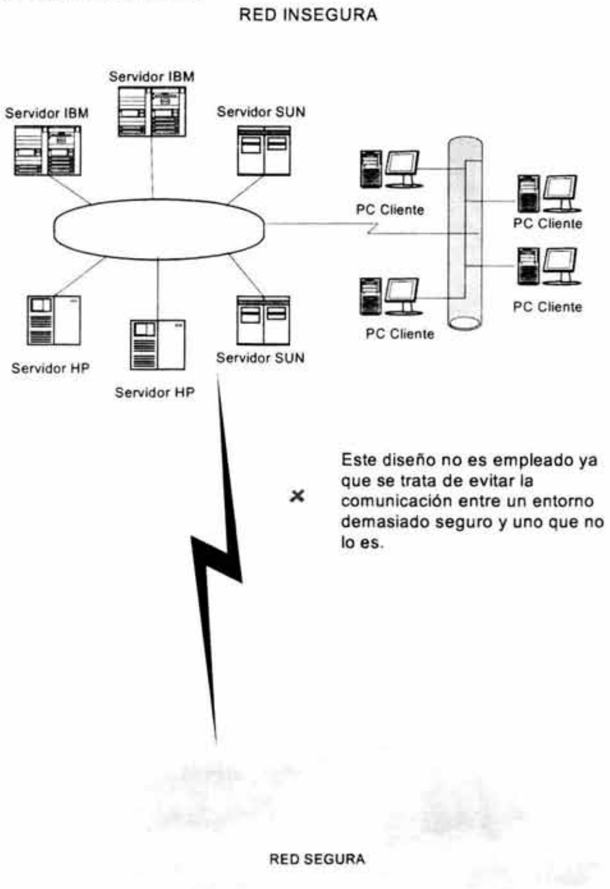


Figura 4.3

Referente a la ubicación física, están en lugares completamente diferentes y protegidos entre ellos a través de firewalls, lo que impide totalmente la comunicación entre ambos entornos. Se decidió en su momento que esta comunicación no era indispensable, pero ahora se considera que no es necesario tal aislamiento, se puede tener un esquema de seguridad en donde se encuentren todos los servidores, sin la necesidad de construir lugares o centros de cómputo independientes y poder tener una administración más controlada; desde ese punto de vista se tendría algo más o menos similar a la figura 4.4 :

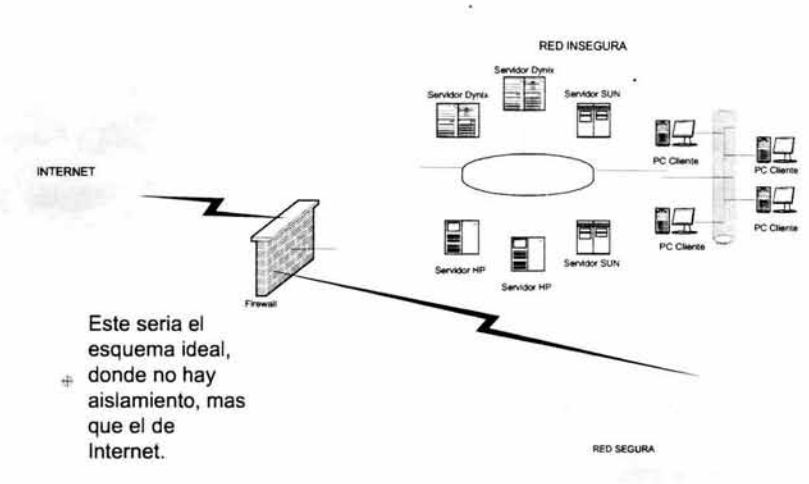


Figura 4.4

4.4.3 Vulnerabilidad en el entorno desprotegido

Se han mencionado algunas vulnerabilidades en el entorno desprotegido y si se desea llegar a lo mostrado en la figura anterior, sólo se logrará teniendo un esquema de seguridad similar al del entorno protegido, pero ese no es el objetivo final, si no tener un mejor esquema, que sea más flexible en su uso, que no consuma muchos recursos de los equipos y por lo tanto, estos se ocupen por otras aplicaciones, que su configuración sea sencilla para adecuarlo a las necesidades solicitadas. También se busca que no quede sólo en esquema de seguridad, si no también llegar a un esquema de monitoreo usando las herramientas de seguridad o scripts que se encarguen de esta tarea, de manera similar (aunque rudimentaria) a los paquetes de monitoreo.

4.4.4 Capacitación

Otro problema que se debe considerar es la capacitación, tal vez sea un poco absurdo ver ésta como tal, pero tiene una justificada explicación; cuando se adquiere software comercial de cualquier tipo regularmente incluye capacitación para su uso, a nivel de usuario y a nivel de administración (si éste lo requiere), esto aparentemente no es un problema, pero viéndolo desde otro punto de vista se encuentran detalles como : esta capacitación solo se le da a un grupo reducido de personas [4], los manuales de uso y administración (suponiendo que es de una herramienta de seguridad) sólo se obtienen asistiendo al curso y estos no se encuentran disponibles en algún sitio de internet, si ocurre algún problema con este software, se requiere que personal de la empresa que lo vendió para su reparación, por que este tipo de detalles no se ven en los cursos y por lo tanto, ocasionan una dependencia hacia las empresas. A diferencia del software de libre distribución que no requiere de capacitación o tal vez si pero de manera autodidacta y muy sencilla, pues hay mucha información en internet acerca de cualquier herramienta libre, en las que explican como usarla y como administrarla. Para detalles de problemas comunes con este tipo de herramientas, solo basta leer un poco la documentación y encontrar la sección de resolución de problemas (throubleshooting). Esta documentación viene generalmente en inglés, debido a que tal vez otros países se han preocupado un poco más por documentar este tipo de material, cosa que no ha sucedido aquí en nuestro país.

Cuando en algún lugar se brinda capacitación, se debe tomar en cuenta que las personas que asistieron a los cursos no siempre estarán trabajando para la empresa o entidad, estas pueden desertar, entonces se tiene que invertir de nuevo en la capacitación del nuevo personal, si estos vuelven a desertar es necesario volver a capacitar a las nuevas adquisiciones y así sucesivamente, cosa que no sucede con las herramientas libres que, como se mencionó anteriormente, todo lo que se necesita saber se puede encontrar en internet.

4.5 ANÁLISIS COSTO-BENEFICIO

El interés inicial de este proyecto se basa en los grandes beneficios que se pueden obtener con las herramientas de libre distribución. A continuación se realizará un análisis costo–beneficio de las herramientas y con esto poder visualizar mejor las características de ambas, además de sus pros y contras.

4.5.1 Estimación del costo-beneficio de las herramientas comerciales.

Este es uno de los puntos esenciales en el análisis, por ser uno de los factores que influyen en la inclinación hacia las herramientas de libre distribución, y el motivo es sencillo, solo basta saber un aproximado del costo del software comercial para descartarlas; el costo aproximado de un elemento de una suite de seguridad rebasa los $ 20,000 dólares, esto es, arriba de $200,000 pesos.

Este precio solo incluye la adquisición del software, se debe considerar que existen otros gastos que se deben realizar al adquirir este tipo de productos, y esto es por que se está contratando un servicio "completo" y que por lo tanto, cada parte del servicio tiene un costo adicional. Estos son gastos elevados a pesar de que traerán beneficios a corto o a mediano plazo.

Otros Costos :

- Otro gasto a considerar es el personal que se debe contratar, debido a que las personas que se encuentran laborando en la empresa o entidad ya tienen definidas sus propias funciones, y esto traerá un gasto adicional que no será temporal, si no permanente, por tratarse de un empleado más, considerando que sólo se contratara uno, pero regularmente se contrata personal extra, por si ocurre algún percance o contrariedad. Otro tipo de personal que se debe

4 No se cuenta con un número aproximado de los usuarios a los que se capacita.

contratar es el de la empresa del software comercial, para la implantación de la suite del software adquirido, aunque solo será durante la fase de implantación (instalación y configuración), pero no deja de ser otro costo.

- La consultoría también tiene un costo extra, ya que cuando termina la fase de implantación, el personal de la empresa del software comercial no tiene obligación de dar asesoría, y éste es otro de los servicios que llegan a vender. Este servicio también es conocido como soporte que se otorga ya sea vía telefónica o en sitio, obviamente el segundo tiene un costo muy elevado, aunque trae mayores beneficios por tener una persona en sitio experta en resolver problemas que se pudieran presentar.

- Otro costo que es muy común que no se tome en cuenta, es el de adquirir software adicional. Es muy normal encontrar software que dependa de otros, en el mayor de los casos de compiladores o si no de algún otro software que permita el funcionamiento de la herramienta instalada.

- La capacitación es indispensable y no se puede descartar al momento de adquirir cualquier tipo de software, ya que de esta depende el aprendizaje del uso y administración de cualquier herramienta de seguridad, para que sea usada debidamente y no sea contraproducente tenerla, esto es, que en lugar de proteger los equipos, se configure de manera incorrecta y perjudique a estos, generando grandes huecos de seguridad [5].

Beneficios

A pesar de que la adquisición de software implica muchos gastos, se obtienen algunos beneficios, que a continuación se mencionan :

Estar a la vanguardia en la actualidad es demasiado importante, sobretodo hablando de software, y si éste es comercial, es más fácil tener la versión más actualizada, ya que inmediatamente que aparece una nueva versión, se le informa al cliente para que la adquiera, el único detalle es que implicaría un costo extra.

Otro de los beneficios es la eliminación o reducción de errores, siempre y cuando se cuente con la capacitación, y además la contratación de personal de soporte en sitio, lo que garantiza que si el problema no es resuelto por el administrador del software, entonces se acude al ingeniero en sitio y en caso de que no se pueda resolver, la última opción sería el soporte vía telefónica, por lo tanto, hay mucha redundancia en la resolución de problemas, y esto es una gran ventaja cuando estos son críticos.

Lo anterior en un ambiente de producción es muy importante ya que agiliza el tiempo de resolución de problemas, esto es, hay pocos tiempos muertos en la producción. Es casi imposible tener trabajando un sistema al 100% los 365 días del año, sin embargo, siempre se procura estar cerca de este porcentaje, ya que esto hace un sistema muy confiable, aunque implique grandes gastos; éste es el objetivo de cualquier empresa o entidad, ya que gastar poco puede traer mayores gastos con la pérdida de tiempos de producción.

5 Esta observación se hace por que existen necesidades de capacitación en aspectos técnicos particulares del software que se adquiere.

Un punto importante dentro de los beneficios es que se contratará gente que se debe especializar en estas herramientas, lo que conlleva a tener gente experta, a pesar de que se debe considerar un gasto permanente, sin embargo, esto ocasionará una excelente calidad en el uso y la administración de todas las herramientas. Una buena administración trae muchas ventajas, como la reducción o ausencia de errores, y por lo tanto, evita la perdida de tiempo en buscar solución a algún error.

La reducción en la adquisición de recursos extras a los contemplados es otro beneficio, por que sólo se tiene un gasto fijo inicial y regularmente nunca hay un gasto posterior. Lo anterior se refiere a que se hace una cotización de todo el software a adquirir y con base a ésta se reserva un gasto (casi siempre elevado) y ya no se vuelve a invertir en ningún software extra.

Tener software actualizado siempre será un gran beneficio, ya que la mayoría de las empresas procuran estar a la vanguardia en hardware y software, entonces es indispensable tener software compatible y esto solo se logra con versiones actualizadas.

Todo lo mencionado en los párrafos anteriores es con el fin de visualizar los beneficios que una empresa o entidad logra, principalmente en la confianza que esto origina en todos los aspectos, como la protección de los equipos, la garantía de un buen servicio, la satisfacción de los usuarios, mayor competitividad y lo más importante la seguridad en los servidores.

4.5.2 Estimación del costo-beneficio de las herramientas de libre distribución.

Para las herramientas de libre distribución es muy difícil sacar un análisis costo-beneficio, basándose en que no se puede estimar el costo de éstas, debido a que son de libre distribución (sin costo), entonces esto se omitirá, sin embargo, el que no tenga un costo implica que tenga algunas desventajas considerables así como grandes beneficios.

Desventajas al adquirir un producto de libre distribución :

- Uno de las principales desventajas es que el administrador del sistema deberá de conocer como trabaja la herramienta a usar, ya que se encargara de su instalación, configuración y administración, esto significa que deberá ser autodidacta hasta que domine perfectamente la herramienta.

- No existirá un soporte para cuando exista una falla, si no que el administrador la deberá resolver, leyendo sus manuales o investigando la causa de la falla, y esto significa emplear tiempo y dedicación hasta que quede resuelta, olvidándose un poco de sus tareas cotidianas, aunque esto se puede ver como una tarea más de administración. Cuando no se tiene mucha experiencia en el uso de la herramienta y se presenta una falla, se reflejara en el tiempo de resolución de ésta, lo que puede llevar algunas horas, tiempo que estará vulnerable el servidor.

- Algunas herramientas necesitan de algún compilador para poder instalarse (como se vió en el capítulo 3), entonces será necesario la instalación de uno o más compiladores, según se requiera. Esto es una desventaja, por que se debe de instalar software adicional, por lo tanto, se debe conocer también su funcionamiento.

- Una gran desventaja es que no existe capacitación para el uso de las herramientas, entonces será necesario aprender a través de manuales, listas de seguridad o en internet, para poder tener un completo dominio de la herramienta a instalar.

Beneficios

A continuación se mencionarán los beneficios que se obtienen con software de libre distribución :

Existen sitios en internet en donde se distribuyen las ultimas versiones de estas herramientas, obviamente de manera gratuita y para evitar que sean alteradas, la mayoría vienen firmadas con MD5. Tener la versión más actualizada dependerá de la constante revisión que realice el administrador.

Con estos productos no existen gastos normales o extras, sólo se requiere de tiempo y dedicación para dominar la herramienta. Esto es demasiado importante en una empresa o entidad, ya que es muy difícil que autoricen la compra de algún software, principalmente si este llega a ser demasiado costoso, entonces se convierte en extraordinaria la opción de adquirir un software gratuito con la misma funcionalidad que el comercial y en ocasiones superando a éste.

Otro beneficio es que a pesar que sea software de libre distribución, existe compatibilidad con versiones comerciales, entonces no hay peligro de quedar obsoletos, y como se mencionó anteriormente, dependerá del administrador siempre contar con las versiones mas actualizadas.

En lo que se refiere a funcionalidad, los de libre distribución son mucho más confiables, por su flexibilidad en la configuración y, por lo tanto, mayor seguridad en los servidores en donde queden instaladas estas herramientas.

4.6 MÉTODO DE TRABAJO

El método de trabajo para la propuesta de integrar uno o más esquemas de seguridad, utilizando herramientas de libre distribución dentro de esta entidad gubernamental, se lleva a cabo de la siguiente manera :

4.6.1 Parámetros de estudio

Para poder comenzar a hablar sobre un tema, es necesario conocer lo básico de éste, para este caso el tema principal es la seguridad en Unix que va de la mano con la administración de este sistema operativo. En los primeros capítulos se detallo acerca de estos 2 temas, la administración de unix básica y avanzada, sobretodo se detallaron puntos que para efecto de esta tesis eran sumamente importantes y esenciales. Esto se debe por que para poder implementar herramientas de seguridad avanzadas era necesario primero cubrir los huecos de seguridad que vienen por default con el sistema operativo Unix, además de, cosas básicas como : los respaldos, políticas que se les deben imponer a los usuarios y una de las cosas que a pesar de ser elemental y sencilla siempre se pasa por alto, que es la revisión de las bitácoras, ya que si se revisaran constantemente, se podrían evitar problemas de todo tipo en cualquier servidor y en todo momento.

En el segundo capítulo se abordó el tema de las características y aplicaciones de las herramientas de seguridad de libre distribución, esto nos servirá de apoyo en el último capítulo, en donde se evaluará que herramienta se utilizará en cada esquema de seguridad, se podrán usar todas o solo algunas, aunque también depende de lo que se requiera, por que no necesariamente todas las herramientas tienen que ser usadas al mismo tiempo o en el mismo servidor, si no que su uso depende de las necesidades de la entidad, empresa o inclusive del área laboral. Entonces este capítulo fue una introducción a las herramientas de seguridad y para conocer el objetivo de estas en el ambiente computacional. Como el tema de la seguridad es demasiado extenso se mostraron diferentes tipos de herramientas, como :

Autentificación : Estas nos sirven para obligar al usuario a tener o usar contraseñas que no sean débiles.

Cifrado de datos : Estas se encargan de cifrar los datos que viajan por la red, antes de que estos sean enviados.

Comunicación : Estas son las más populares y por lo tanto más usadas, ya que son muy efectivas por que establecen comunicaciones seguras entre el cliente y el servidor.

Monitoreo de red : Estas se pueden catalogar como analizadoras o preventivas, por que ayudan a evitar problemas de seguridad en un servidor, además de que en caso de un ataque se pueden usar como auxiliares para detectar de donde proviene éste y que es lo que se está enviando.

Monitoreo del sistema : Estas son muy usadas para poder detectar los problemas de seguridad que tiene un servidor, ya que verifican los probables huecos que existan.

Otras herramientas : También se vieron otras herramientas que son auxiliares en la seguridad del sistema, y que para el caso de esta tesis resultarán muy útiles.

4.6.2 Mecanismos de evaluación para otorgar una solución

En otro capítulo se instalaron y configuraron cada una de las herramientas de libre distribución, para verificar su funcionamiento y detalles que se fueran presentando, los cuales regularmente no vienen documentados. En esta tesis se mencionaron algunos tips, para que cuando un administrador tenga problemas con estas herramientas, no pierda tiempo en la resolución de estos y solo se guié por las recomendaciones aquí mencionadas. También se ejecutaron pruebas de funcionamiento de estas herramientas, para que en la fase de implantación de los esquemas de seguridad que se van a proponer, no se presente algún problema al instalarlas o configurarlas o para saber si requiere de algún software complementario y analizar si se puede adquirir éste o si es factible tenerlo, es por esto que se realizó todo este análisis previo a la implantación. En el capítulo en donde se implantarán los esquemas de seguridad sólo se verá algo de configuración, se pondrá más atención en los esquemas en general y en sus beneficios, todas las pruebas se realizaron en el capítulo 3.

En algunos esquemas será necesario configurar algunas cosas adicionales, para adecuar las herramientas a conveniencia de estos y por necesidades de la entidad gubernamental.

4.6.3 Análisis de la situación

En este capítulo se vio un análisis de la situación actual de la entidad gubernamental, empezando por un resumen general de lo que es la seguridad y sus principales objetivos, partiendo de las bases de los primeros capítulos, o sea, administración y seguridad del sistema operativo unix, con esto se tiene un amplio panorama del principal problema que aqueja a la entidad. Después se hizo un análisis a detalle de otros elementos que perjudican o pueden perjudicar a la entidad, como puede ser desde un usuario, la persona o personas que administran los servidores por una mala administración o inclusive por alguna persona malintencionada que busque afectar a la entidad, utilizando diferentes medios para conseguirlo.

Otro punto que se tocó en este análisis fue la definición de por que se consideraba un problema, contemplando puntos como limitaciones en los servidores, en los usuarios o inclusive en los administradores, la ubicación de los entornos de servidores con los que cuenta esta dependencia, explicando por que se considera que la ubicación está mal, debido a la protección que tiene uno y lo desprotegido que está el otro, lo que origina que este ultimo tenga el problema que es sumamente vulnerable a algún ataque sofisticado.

La capacitación es un punto importante que se analizó, ya que si se lleva a cabo de una manera adecuada, se pueden obtener grandes beneficios, por que no sólo es capacitar por capacitar, si no saber a que personas y en que lo van a aplicar a corto, mediano y largo plazo.

También se sacó un análisis costo-beneficio de las herramientas de seguridad comerciales comparadas con las herramientas de seguridad libre distribución, con el fin de obtener los beneficios de cada una y determinando por que la inclinación hacia las herramientas de seguridad libre distribución, pero sin olvidar que pueden usarse ambas, aunque el objetivo es más que claro, al querer mostrar que la mejor opción es usar las de libre distribución.

5. CAPÍTULO 5: PROPUESTA DE INTEGRACIÓN DE UN ESQUEMA DE SEGURIDAD EN EL CENTRO DE CÓMPUTO DE LA ENTIDAD GUBERNAMENTAL, UTILIZANDO HERRAMIENTAS DE LIBRE DISTRIBUCIÓN.

5.1 INTRODUCCIÓN

En los primeros capítulos se analizaron elementos básicos y avanzados de la administración del sistema operativo Unix, también se tocaron temas acerca de la seguridad de este sistema operativo, como mejorarla e implantarla de manera básica. Para lograr una protección más avanzada en el sistema operativo Unix, en capítulos posteriores se vieron características y funcionamiento de herramientas de seguridad, así como su instalación y configuración de éstas.

Después de terminar con los temas elementales, en el capítulo anterior se vio la problemática actual que tiene la entidad gubernamental, para tener un panorama más amplio de lo que se busca solucionar y así, en este capítulo poder determinar cual es la mejor solución a implantar, pero solo para este caso en particular, ya que el objetivo de esta tesis no es únicamente dar una solución para esta entidad, si no , emplear cualquiera de los esquemas que se van a proponer en cualquier lugar, ya sea en una empresa privada, entidad gubernamental o inclusive en instituciones educativas.

En este capítulo se van a proponer tres esquemas de seguridad usando las herramientas vistas en los capítulos anteriores; al final se seleccionara una que se acople a las necesidades de esta entidad gubernamental, tomando en cuenta que no se debe afectar el desempeño de los servidores y tratando de igualar o mejorar lo que se ha implantado con las herramientas comerciales, pero sobretodo, el objetivo primordial es proteger la información que se encuentra vulnerable ante un ataque malintencionado.

5.2 CONSIDERACIONES DE SEGURIDAD PARA IMPLANTAR UN NUEVO ESQUEMA.

Antes de proponer los esquemas de seguridad, se deben tener algunas consideraciones importantes, las cuales deberán emplearse en cualquier lugar en donde se llegará a implantar un esquema, a continuación se mencionarán algunas:

5.2.1 Definición de los servidores que serán configurados.

Lo primero que se debe conocer, es la disponibilidad de equipos en donde se hará la implantación, esto es, los equipos en donde se pueda armar un escenario de instalación y configuración, y de ser posible tener un ambiente de pruebas, en donde se pueda hacer todo lo anterior antes de pasarlo a un ambiente de producción. En este caso se cuenta con servidores de Sun, IBM y HP, esto es que se cuenta con servidores de diferentes plataformas, pero con el sistema operativo similar, al fin y al cabo Unix.

Cada uno de los servidores tiene funciones específicas e independientes (figura 5.1) , ya que cada plataforma trabaja de manera aislada, cada una con diferentes versiones de software de aplicación, pero con ambientes de trabajo similares, lo anterior significa que cada uno de los servidores de las distintas plataformas cuentan con : la parte de desarrollo de aplicaciones, ambiente de pruebas, ejecución de respaldos, distintas bases de datos y un ambiente especial para la producción, siendo éste el más crítico. Por el momento no se cuenta con un servidor Web, pero se tiene contemplado tenerlo a corto plazo, asignándolo en alguno de los servidores existentes, lo que implicaría una aplicación adicional y a su vez otros huecos de seguridad.

Figura 5.1 FUNCIONES DE CADA PLATAFORMA

5.2.2 Seguridad a nivel de Unix en los servidores en donde serán instaladas las herramientas.

Antes de empezar con la selección de las herramientas a instalar, se deberá hacer una revisión de estos servidores a nivel de sistema operativo, siguiendo los siguientes pasos :

- Asegurarse de que el archivo /etc/passwd esté con los permisos adecuados y tener activado el /etc/shadow.
- Antes de comenzar a aplicar herramientas, hacer una revisión de los permisos (suid, sgid, sticky bit) en archivos que son críticos para el sistema.
- Verificar que no existan archivos que permitan conexiones remotas, tal es el caso de : /etc/hosts.equiv y .rhosts
- Revisar si existen listas de control de acceso.
- La funcionalidad en los servicios de red es demasiado importante, entonces estos deben estar configurados correctamente. Algunas herramientas tienen que acceder a otros servidores, por lo tanto, es necesario saber que servicios de red se van a abrir y cuales se cerrarán.
- Deshabilitar los servicios de red innecesarios, que pudieran permitir algún ataque debido a su vulnerabilidad.
- Habilitar el uso de terminal segura, hasta que queden configuradas las herramientas de seguridad y nos permitan conexiones remotas seguras.
- Revisar las cuotas asignadas a los usuarios y con esto evitar que se puedan llenar los Filesystems, en donde se vayan a alojar las bitácoras de las herramientas.
- También es necesario revisar que parches están instalados en los servidores; es conveniente aplicar los parches más actuales, los cuales se encuentran disponibles en sus respectivos sitios de Internet (sun, ibm y hp).
- Revisar que tipo de auditoria tiene el sistema y de donde está obteniendo información para sus bitácoras, debido a que algunas herramientas que serán instaladas, requieren de estos datos.
- Otro punto importante que no debe pasar desapercibido, es que antes que se instale cualquier tipo de herramienta, se debe obtener un respaldo completo del servidor, y conforme se vayan instalando éstas, se deberán respaldar los archivos o directorios que se irán modificando.
- Se recomienda cambiarle los permisos a los siguientes comandos, por que su uso principal es de administración y no para los usuarios normales. A todos estos comandos se les debe de cambiar los permisos para que solo root los pueda ejecutar o si es sumamente necesario, otorgarle los permisos a otros usuarios a través de un grupo especial, que se creé para este fin. Estos son de los comandos más importantes y riesgosos :

```
/bin/su
/etc/netstat
/etc/ping
/etc/traceroute
/usr/bin/crontab
/usr/bin/rcp
/usr/bin/resh
/usr/bin/rlogin
/bin/ipcs
/bin/mail
/bin/rmail
/bin/write
/etc/dmesg
/etc/showcfg
/etc/wall
/usr/bin/mailx
```

Después de haber realizado los pasos anteriores, ahora se procederá a definir las herramientas que serán utilizadas en el diseño de los esquemas.

5.2.3 Definición de las herramientas de seguridad.

A continuación se hará una breve sinopsis de cada una de las herramientas:

Npasswd

Esta herramienta será de gran apoyo para el administrador, para evitar que el usuario tenga contraseñas débiles, sin la necesidad de capacitar ampliamente al usuario, si no, solamente con enseñarle a usar un comando "más" del Sistema operativo Unix.

Pgp

Esta herramienta de cifrado es de gran utilidad cuando no se puede tener comunicaciones seguras por medio de programas como el secure shell. En ocasiones por "x" causa, es necesario utilizar programas como el telnet o el ftp, los cuales no son nada fiables, entonces se requiere de una herramienta auxiliar como PGP, que cifra la información antes de enviarse y sólo la puede descifrar el destinatario.

Secure shell y Openssh

Estas 2 herramientas son las más usadas, por la facilidad de su uso, y todas las ventajas que trae, como comunicación segura a través de una terminal (telnet seguro), transferencia de archivos confiable (ftp seguro), despliegue de gráficos a través de un tunneling seguro (emulador gráfico), todo esto usando solo un puerto y de manera segura.

Tcp-wrappers

Esta herramienta sirve como un guardián de las peticiones que se hacen a los servicios de red de un servidor. También analiza todo lo que se recibe, a parte de guardar esta información en bitácoras, para su posterior revisión. Otra de sus características importantes es que puede realizarse un filtrado de direcciones Ip, para denegar u otorgar ciertos servicios de red solamente.

Sniffer (como herramienta analizadora)

Esta no es como tal una herramienta de seguridad, si no solamente es una herramienta auxiliar que sirve como analizadora de lo que fluye por la red y con ella se pueden empezar a descartar puntos de falla, que estén afectando a algún servidor.

Portsentry

Otra herramienta que sirve de guardián es la de porsentry, la cual se va a encargar de proteger al servidor de cuando le realicen "scanneos" o "barridos" de puertos, para saber la disponibilidad de ellos e intentar un ataque por los que se encuentren abiertos. Entonces el objetivo de esta herramienta es detectar estos "scanneos" y evitar que lo siga intentando.

Cops

Esta herramienta es para monitorear el servidor y se encargara solamente de detectar vulnerabilidades, sin embargo, solamente es preventiva más no correctiva, dependerá de la elaboración de shells scripts para la corrección de los problemas que encuentre la herramienta.

Tripwire

Otra herramienta para monitorear el sistema es la de tripwire, y sirve para obtener un "snapshot" del sistema en cierto momento y hacer una comparación de éste cada determinado tiempo y poder detectar cambios en archivos críticos para el sistema. Esta se usa principalmente cuando no hay una administración centralizada y el servidor es administrado por varias personas, esto es, sirve como manejador de control de cambios a archivos críticos del sistema.

Saint

La herramienta de saint es similar a las 2 anteriores, ya que está diseñada para detectar huecos de seguridad en un sistema, sin embargo, tiene cosas adicionales como : se instala solamente en un servidor, y éste se encarga de analizar equipos remotos, además recolecta información de hardware y de software, entonces se puede usar para sacar inventarios y puede ser ejecutado por cualquier usuario sin tener privilegios de root.

Cabe mencionar y recomendar que se debe de actualizar esta herramienta cada que haya una nueva versión, por que en su base de datos traerá los huecos de seguridad que vayan encontrando, ya sea a nivel de sistema operativo o de aplicaciones.

Sudo

Esta herramienta adicional es de gran utilidad cuando se tiene más de un administrador, ya que permite la restricción en el uso de comandos y se tiene un buen control en la administración de uno o más servidores.

Nullshell

Por último la herramienta de nullshell es de utilidad para desactivar cuentas de Unix, sin la necesidad de eliminarlas, ni de borrar sus archivos y estructura. Cuando se investiga acerca de un ataque o de intentos de conexión mediante una cuenta caducada, esta herramienta registra estos intentos y los almacena en una bitácora, para tener un reporte completo y detallado de lo que está pasando.

5.2.4 Manejo de permisos en dispositivos y comandos.

Debido a que en la actualidad, en algunos lugares aun se trabaja con equipo obsoleto, no es posible instalar herramientas como sudo, pero esto no quiere decir que estos servidores quedarán desprotegidos, si no, se deben de buscar otras opciones. Para el caso de los respaldos, se optará por manejarlo a través de permisos en los dispositivos en donde se respalda la información. En la mayoría de los unix, por default los permisos de los dispositivos son de escritura para todos los usuarios del sistema, esto es un gran peligro, por ejemplo si el usuario root lanza un respaldo e inmediatamente que termina éste, cualquier usuario lanza otro respaldo al mismo dispositivo, se sobrescribirá lo que hay en la cinta. Para evitar esto, se puede crear un grupo especial o asignar al usuario que realizara los respaldos al mismo grupo que el dispositivo y con esto solo los miembros de este grupo podrán utilizar el dispositivo (unidad de cinta).

En caso de contar con la herramienta de sudo, se recomienda que los dispositivos tengan permisos para que solo root pueda lanzar respaldos, y como sudo adquiere privilegios de root en ciertos comandos, sólo los usuarios que estén permitidos podrán respaldar, en este caso será un usuario especial que ejecute solo la tarea de respaldar.

Lo anterior también se puede aplicar principalmente para comandos de administración y que no deben de ser usados por usuarios normales; entonces se les otorga permisos para que solo root o un selecto grupo de usuarios puedan ejecutarlos. Algunos ejemplos comandos en los que se debe de tener cuidado son : finger, rlogin, rsh, rcp, talk, write, netstat, ftp, telnet, su, ifconfig, route, ping y traceroute, por mencionar algunos. Esto dependerá mucho de la funcionalidad que tienen los servidores, por que tal vez unos comandos deberán ser usados por los usuarios normales.

5.2.5 Shells de monitoreo

También se busca que esta tesis no solamente se enfoque en esquemas de seguridad, si no también tener un esquema de monitoreo usando las herramientas de seguridad o scripts que se encarguen de esta tarea, de manera similar (aunque rudimentaria) al software comercial de monitoreo.

Se pueden tener 2 tipos de monitoreo, ya sea por medio de un equipo central que este monitoreando a los demás servidores o de manera local en caso de que no se cuente con equipos adicionales. Para el caso de monitoreo remoto se emplearán scripts muy básicos, dependerá de cada administrador que tan complejo o detallado quiera realizarlos. Aquí sólo se propondrá que se puede monitorear.

5.2.6 Monitoreo remoto

Como se mencionó en el párrafo anterior el monitoreo remoto se llevará a cabo desde un servidor central, el cual empleará los siguientes comandos :

ping : Este comando es el más utilizado cuando existen problemas de comunicación, es al que se recurre antes que cualquier otro, entonces la idea es de tener un script que se ejecute en breves lapsos de tiempo, para verificar que respondan todos los servidores y en caso de que alguno no lo haga, inmediatamente verificar el punto de falla. Tal vez podría avisar con un mensaje en pantalla, indicando que servidor no está respondiendo.

tnsping *:* Este comando es muy similar al anterior, solo que se emplea para el monitoreo de base de datos de manera remota, lo cual es muy útil cuando se tiene más de una base de datos y en diferentes servidores.

telnet al puerto 80 *:* Se empleara el telnet como un auxiliar para monitorear el puerto del servidor web, esto es, que responda a cualquier petición. Este monitoreo es muy básico, si se quiere algo más avanzado, dependerá de que servidor web se esta utilizando y tal vez emplear herramientas de éste. El hecho de que no responda el telnet por el puerto 80, puede ser suficiente para determinar si se envía un mensaje en pantalla y revisar la aplicación.

5.2.7 Monitoreo local

Algunos comandos solo se pueden utilizar de manera local, y enfocados a monitorear el performance y uso del servidor, a continuación se darán algunos comandos que pueden ser de gran utilidad :

Rendimiento de CPU

Uno de los factores que afecta en gran medida el rendimiento de cualquier servidor es el uso del procesador, entonces se debe tener un monitoreo constante a éste y de preferencia guardar datos históricos de su comportamiento, para que se puedan sacar estadísticas o para determinar que puede estar afectando el rendimiento del CPU. El comando que se emplea para obtener información es el sar :

```
server1:>sar  1  5

20:18:31   %usr   %sys   %wio   %idle
20:18:32    28     29     0      44
20:18:33    17     41     1      42
20:18:34    21     45     0      34
20:18:35    34     55     0      11
20:18:36    19     38     0      43

Average     24     41     0      35
```

En donde :

```
server1:> sar  x  y
```

x	Intervalo de segundos en que aparece
y	Cuantas veces se va a ejecutar
%usr	porcentaje que están ocupando los usuarios.
%sys	porcentaje que están ocupando algunas aplicaciones del sistema.
%wio	Este se refiere a procesos de I/O o de memoria (swap/paginación).
%idle	Otros procesos.

Memoria utilizada

También se debe tomar en cuenta la memoria que se está ocupando, aunque no existe un comando como tal para verificarla, sin embargo, se pueden usar comandos auxiliares, como el vmstat, del cual se puede obtener información para saber la memoria RAM real existente y la memoria RAM utilizada. Aquí se muestra un ejemplo de cómo obtenerla :

```
server1:> vmstat
     procs        memory              page                      faults     cpu
  r   b   w    avm   free  re  at  pi  po  fr  de  sr  in   sy    cs  us sy id
  1   0   0  21282 217571 119  26   0   0  34   0   0 419 14645  657  21 41 38
```

La parte sombreada indica valores de la memoria, de aquí se puede obtener a través de cálculos matemáticos los valores de la memoria RAM total y la utilizada..

Swap utilizada

El comando para verificar el swap, varia para cada unix, sin embargo en todos existe por lo menos uno, algunos de los comandos existentes son : swapinfo y swap, con sus diferentes opciones. Este es de gran ayuda para saber la disponibilidad de la memoria swap, ya que también le pega al performance, y de ser así, solo es necesario incrementarle más memoria.

Espacio en disco

Otra variante que se debe monitorear es el espacio en los filesystems, aunque parezca algo muy básico y muy trivial, el tener un monitoreo constante al espacio almacenado, permite prevenir muchos problemas e inclusive evita que se caiga el servidor en caso de llenarse el filesystem de root. Este se hace a través de comandos como el df y el bdf.

Conexiones exitosas (bitácoras)

Es conveniente realizar una verificación constante a las bitácoras, que son las principales fuentes de información de los problemas que presente un servidor y además aquí están de manera detallada todo lo que ha sucedido con el servidor, como intentos de conexión al servidor, cuando se da de baja y cuando se inicia el servidor, fallas en dispositivos (hardware), problemas con la memoria, problemas con parámetros del kernel, etc., entonces solo basta emplear algunas utilerías de unix para facilitar la tarea de revisión, ya que son archivos demasiado grandes y no es factible verlos con un editor de texto, entonces bastará con un filtrado de palabras claves sobre la bitácora, cada determinado tiempo, sin que sea muy constante.

Base de datos

Por último, también se debe tener un monitoreo constante a las bases de datos, en caso de que existan. Aquí basta con una conexión a la base de datos para saber si responde o no, con esto será suficiente para aislar de manera inmediata problemas con la información almacenada.

5.3 PROPUESTA DE 3 ESQUEMAS DE SEGURIDAD PARA IMPLANTAR EN LA ENTIDAD GUBERNAMENTAL.

Antes de comenzar con la propuesta de los esquemas, se deberá recordar que los servidores que se mencionaron anteriormente (IBM, Sun y HP), son solo equipos muestra y se utilizarán en cada esquema sólo para que estos sean más entendibles, y cuando se realice la implementación en cualquier empresa o entidad, se sustituya cada servidor (IBM, Sun o HP) por cualquier otro, no importando cual sea su plataforma, siempre y cuando cuente con sistema operativo Unix. Por ejemplo, el servidor IBM que aquí se usará, es un equipo obsoleto y muy antiguo, esto se hace con la intención de que si se llegasen a encontrar con este caso, no sea una limitante al momento de implantar cualquier esquema sugerido en esta tesis, ya que uno de los objetivos principal es tener seguridad en cualquier tipo de servidor Unix.

5.3.1 Definición de 3 esquemas de seguridad

Durante los 4 capítulos anteriores se fueron dando bases para poder llegar hasta este último capítulo, en donde ya se definirán los esquemas de seguridad recomendados y para propósito de esta tesis, se seleccionara el más adecuado, acorde a las necesidades de la entidad gubernamental. Cada uno de los esquemas que se recomendarán, será con base a 3 plataformas diferentes y una de ellas con limitantes en cuanto a hardware y software, por que desafortunadamente en todas las empresas y entidades, no se puede tener tecnología de punta, por esta razón, se toma en cuenta que pueda existir al menos un servidor obsoleto. A continuación se definirá cada uno de los esquemas.

5.3.2 Esquema 1

El primer esquema (figura 5.2) es el menos complejo de los 3, ya que la configuración será implantada de manera independiente en cada servidor, esto es, se tratará de asegurar de manera robusta cada uno de los servidores.

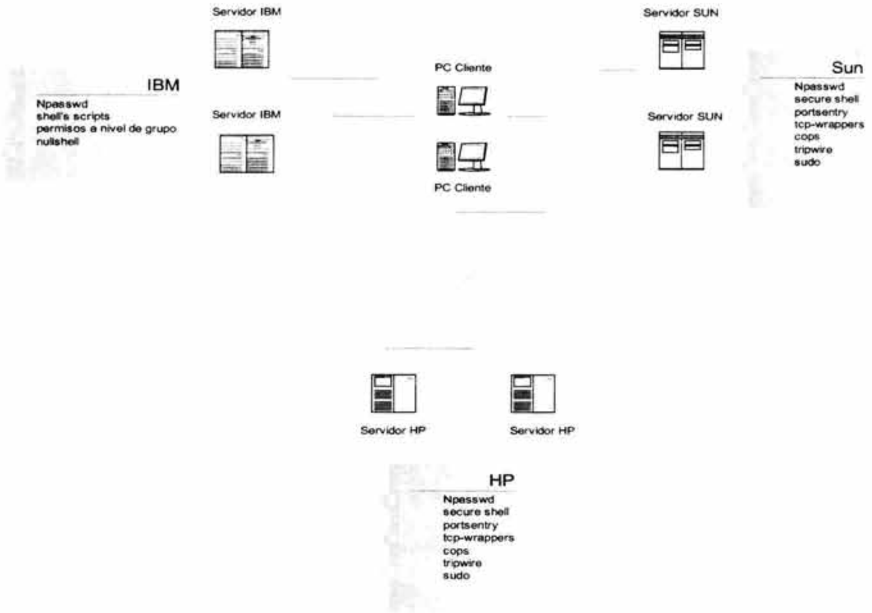


Figura 5.2 PRIMER ESQUEMA PROPUESTO

Este es el primer esquema que se propone para la implantación, aparentemente es el más "sencillo" de configurar [1]. Lo anterior significa que no hay un equipo central que lleve el control de la seguridad, si no cada servidor tendrá sus propias herramientas instaladas, y con esto se tiene una seguridad muy robusta, pero una administración muy descentralizada e independiente, lo que origina revisión de funcionalidad y bitácoras en cada uno de los servidores en donde se tengan instaladas las herramientas, esto es mucho tiempo invertido en estas tareas.

Elementos del esquema # 1

En la figura anterior se muestran los elementos que conforman el primer esquema, como se puede ver, se cuenta con servidores de diferente plataforma, sin embargo, como son independientes unos de otros, la implantación es más sencilla. En este primer esquema se respetará totalmente la distribución de las aplicaciones de los servidores, no se realizará ningún cambio, solo la implantación de herramientas de seguridad. A pesar de que no es el esquema más apropiado ni recomendable, es el más utilizado en cualquier lugar en donde se requiera seguridad y esto se debe a diferentes motivos, el más destacado es la idea errónea de que si un ambiente está funcionando bien, entonces no es conveniente modificarle nada, solo elementos básicos y sin que afecten el funcionamiento actual. Entonces no se puede descartar proponer este esquema, que a pesar de ser sencillo, no deja de ser muy útil.

El objetivo de este esquema es tener servidores muy seguros, entonces lo más viable sería instalar todas las herramientas en cada uno de los servidores, sin embargo, esto

1 A pesar de ser el esquema mas sencillo de configurar, se le instalan mas herramientas que a cualquier otro esquema propuesto

no es necesario y en algunos casos no es posible por las diferentes plataformas que se están manejando

Los servidores IBM no tendrán instaladas las mismas herramientas que los servidores Sun y HP, debido a que los servidores IBM son más antiguos y, por lo tanto, no se les puede actualizar el software ni el hardware, entonces no es posible tener compiladores para instalar las herramientas y mucho menos espacio para las bitácoras que generen éstas.

Plataforma IBM

En este esquema al servidor IBM se le instalarán solamente algunas herramientas, puesto que, como se había mencionado anteriormente, es un servidor obsoleto al cual no se le pueden instalar todas las herramientas aquí mencionadas, entonces será necesario buscar la manera de poder tener un ambiente seguro y que pueda seguir trabajando de manera normal.

Algunas herramientas indispensables en la seguridad de un servidor, no podrán ser instaladas en el servidor IBM, tal es el caso de : secure shell, tcp-wrappers, cops, tripwire y sudo, sin embargo a través de otros métodos se compensará la falta de estas herramientas.

Las herramientas que se le instalarán al servidor IBM son : npasswd, nullshell, shell's scripts y permisos en dispositivos y comandos.

Plataforma Sun

El otro servidor es un Sun, al cual se le pueden instalar mas herramientas que al servidor anterior, aunque también se protegerá de manera independiente, pero al igual contará con un ambiente seguro sin que afecte su funcionamiento. Con este servidor no habrá problemas de instalación y configuración, por que no es un servidor obsoleto como el servidor IBM, es más actual y de mayor capacidad a nivel de hardware y software.

Se instalarán herramientas indispensables en este servidor, tal es el caso de : npasswd, secure shell, portsentry, tcp-wrappers, cops, tripwire y sudo, además de los shell's de monitoreo local.

Plataforma HP

Éste es el tercer y ultimo servidor al que se le instalarán y configurarán las herramientas de libre distribución. Con este servidor no habrá problemas por ser el más actual en cuanto a software y hardware. También se protegerá de manera segura e independiente, no va a interactuar con ninguno de los 2 servidores anteriores (IBM y Sun).

En este servidor se instalarán las siguientes herramientas : secure shell, portsentry, tcp-wrappers, cops, tripwire y sudo, además de los shell's de monitoreo local.

5.3.3 Esquema 2

El segundo esquema (figura 5.4), es muy similar al primero respecto a los elementos que lo conforman, la variación radica en el monitoreo extra que se implantará. Las aplicaciones que están instaladas en los servidores quedarán intactas en las plataformas IBM y HP, sin embargo, en la plataforma Sun, se hará un ajuste para que un servidor quede especialmente para el monitoreo y el otro servidor tenga todas las aplicaciones, además de las herramientas de seguridad. Este segundo esquema puede ser utilizado como "trampolín" para pasar a un esquema más complejo, esto es, puede implantarse primero el esquema # 1, el cual no requiere grandes modificaciones en el servidor, después partiendo de este esquema, se puede implantar el segundo, sobretodo si no se cuenta con la experiencia necesaria y por último, como parte opcional, se puede pasar a un esquema más complejo.

Se van a pasar aplicaciones de un Sun a otro, de tal forma que quedará como se muestra en la figura 5.3:

Figura 5.3 ESQUEMA DE APLICACIONES DISTRIBUIDAS EN LOS SERVIDORES SUN

Este esquema también contará con seguridad en todos los servidores, ya que se le instalarán las mismas herramientas que las del primer esquema, la diferencia estará en que tendrá herramientas extras como saint y el sniffer, los cuales servirán como guardianes de los servidores que estarán más desprotegidos, que serán los de plataforma IBM.

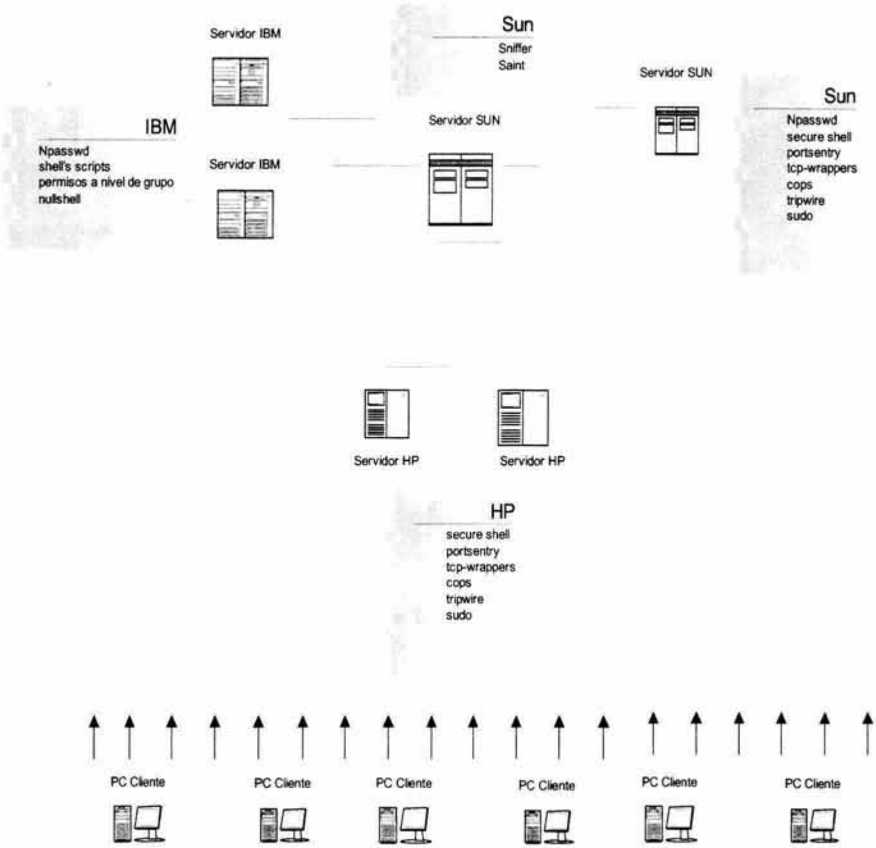


Figura 5.4 SEGUNDO ESQUEMA PROPUESTO

Elementos del esquema # 2

En este segundo esquema se seguirán manejando diferentes plataformas (IBM, Sun y HP), pero estarán distribuidos de diferente manera que el esquema anterior, ahora se nota una distribución un poco más centralizada, debido al monitoreo que se va a implantar, esto implicará que la configuración sea un poco más compleja. En este esquema se distribuirán nada mas las aplicaciones del servidor Sun, las de IBM y las de HP quedarán intactas, como se nota en la figura anterior. Este esquema es más recomendable que el anterior, por tener más seguridad, la desventaja es que se debe contar con un servidor exclusivamente para monitorear.

En este esquema habrá herramientas de seguridad locales, además de las herramientas remotas, por lo tanto, será un esquema con mayor seguridad que el anterior. Las herramientas de monitoreo remotas serán las siguientes : saint, que verificará huecos de seguridad en servidores remotos y un sniffer encargado de monitorear los paquetes que viajan a través de la red.

Plataforma IBM

Lo único que cambiará en todos los esquemas será su distribución y las herramientas de seguridad que se vayan a instalar, por que los servidores seguirán siendo los mismos, en este caso debido a las características de los servidores IBM, se les instalarán pocas herramientas (las mismas del esquema # 1). Entonces no se le podrán instalar herramientas como : secure shell, tcp-wrappers, cops, tripwire y sudo, pero si contará con shell's scripts que compensen la ausencia de éstas, además de las herramientas que si podrán ser instaladas como npasswd y nullshell.

Una de las grandes ventajas de este esquema, es el monitoreo y protección que se le dará al servidor IBM y por ser el que más carencias tiene por ser obsoleto, habrá un sniffer vigilándolo de manera permanente.

Plataforma SUN

En este esquema los servidores Sun juegan un papel relevante, por que no solo tendrán seguridad, si no uno de ellos será el encargado de monitorear el tráfico que pase entre los servidores y también de estar verificando probables huecos de seguridad que puedan tener. Se escogió el servidor Sun por ciertos motivos, entre los que destacan que no es un servidor obsoleto, cuenta con buena capacidad en memoria y procesador, uno de los servidores puede tener todas las aplicaciones y el otro se puede encargar del monitoreo y la más importante, que esta plataforma trae un sniffer propio (snoop). Existirá un servidor de monitoreo y otro servidor que tendrá las aplicaciones y herramientas de seguridad que lo protegerán.

Servidor de monitoreo

Al servidor de monitoreo solamente se le instalará la herramienta de seguridad Saint, además de contar con el sniffer. Debido al diseño de este esquema, es difícil tener un buen control en aspectos de seguridad, entonces esta herramienta será configurada de tal manera que pueda monitorear y analizar lo que fluye entre los servidores más comprometidos, en este caso, los servidores IBM. Se puede tener un monitoreo permanente para vigilar las conexiones a este servidor y que es lo que está realizando cada uno de los usuarios, en caso de que existan muchos usuarios, se puede hacer un filtrado para vigilar solo programas o comandos que pongan en riesgo la integridad del servidor. Todo lo anterior se realiza cuando se comienza a utilizar esta herramienta (sniffer), después de hacer un análisis, se va descartando lo que no tiene caso monitorear y sólo se vigila lo que se considera riesgoso.

Con la herramienta saint se detectará de manera centralizada los probables huecos de seguridad de cada servidor. No es necesario ejecutar esta herramienta con la cuenta de superusuario, ni tampoco abrir servicios remotos en cada servidor (.rhosts, remsh, rexec o rlogin), funciona como cualquier otra aplicación. Otra ventaja de Saint, es que recolecta información de los servidores que se están administrando, y asi poder llevar un control de inventario. Será necesario ejecutarla de manera constante para detectar vulnerabilidades y después corregir estas a través de shell script's o con intervención directa del superusuario.

Otra ventaja es que se pueden entregar reportes (en formato html) de lo que se haya detectado e inclusive saint hace recomendaciones de cómo resolver los huecos de seguridad que encuentre, esto es muy útil, en el caso de administradores inexpertos. Se pueden obtener reportes para entregarlos a los directivos o simplemente para tener documentos históricos de la seguridad de los servidores y llevar un control de cambios que se vayan realizando.

Servidor de aplicaciones

El servidor de aplicaciones tendrá instaladas las mismas herramientas que el esquema anterior : npasswd, secure shell, portsentry, tcp-wrappers, cops, tripwire y sudo, además de los shell's de monitoreo locales, para que genere información del comportamiento del servidor.

Plataforma HP

En este esquema, el servidor HP tendrá instaladas y configuradas las herramientas de libre distribución, igual que en el primer esquema. Este servidor contará con la misma seguridad, sólo que no será tan independiente por que va a interactuar con el servidor de monitoreo, aunque esta parte será transparente para el servidor HP, ya que no habrá cambios en su configuración original.

En este servidor se instalarán las siguientes herramientas : secure shell, portsentry, tcp-wrappers, cops, tripwire y sudo, además de los shell's de monitoreo local.

5.3.4 Esquema 3

En el esquema # 3 se propone un diseño basado en monitoreo centralizado y distribución de aplicaciones en las tres plataformas (figura 5.5), esto quiere decir, que se ocupará un servidor como en el esquema anterior para monitorear los servidores que estén más desprotegidos y además se distribuirán las aplicaciones de cada plataforma, con el propósito de que también se distribuya la instalación de las herramientas de seguridad, para mejorar el performance de cada servidor y la revisión de sus bitácoras. De los tres esquemas propuestos, éste se considera el más complejo, por que las herramientas se deben configurar de acuerdo a la distribución de aplicaciones existentes y las que puedan surgir en un futuro.

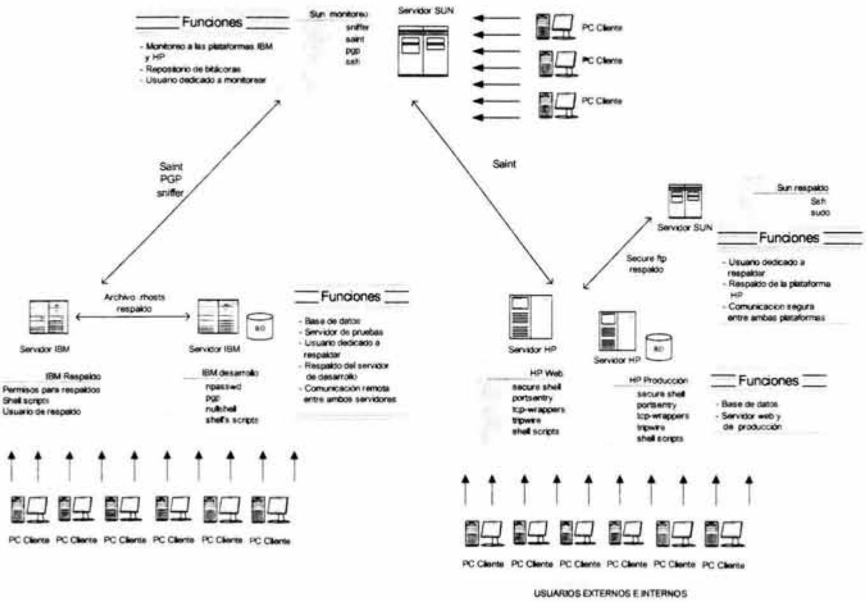


Figura 5.5 TERCER ESQUEMA PROPUESTO

Este esquema aparentemente es el más desprotegido, por que no se instalarán todas las herramientas propuestas en los servidores, si no, se distribuirán conforme a las necesidades de cada uno y tomando en cuenta en que se utilizan. Todos los servidores tendrán una funcionalidad específica, entre las que destacan : la realización de respaldos, el desarrollo de aplicaciones, otro que se ocupe exclusivamente para la producción, probablemente uno que se utilice sólo como servidor Web y, finalmente, si es posible tener un servidor solamente para un ambiente de pruebas.

El servidor central de monitoreo, enfocara su atención en vigilar la plataforma IBM, pero también servirá para tener un monitoreo centralizado hacia los servidores restantes y como repositorio de bitácoras de cada uno de estos, para poder analizarlas y determinar probables fallas que puedan presentarse.

Elementos del esquema # 3

En este último esquema, las aplicaciones y las herramientas se distribuirán de tal forma que exista un monitoreo centralizado, además de que esto beneficiará a los servidores por que ya no van a tener tanta carga de trabajo. Respecto a la distribución de aplicaciones, se va a realizar en cada uno de los servidores (IBM, Sun y HP) y el monitoreo se va a realizar igual que en el esquema anterior, sólo que con elementos adicionales como el almacenamiento de todas las bitácoras.

De los tres esquemas propuestos, éste se puede considerar como el más complejo en lo que se refiere a configuración, pero es el más seguro y confiable, por el tipo de tecnología de los servidores. La selección de las herramientas de seguridad junto con la distribución de aplicaciones serán las partes esenciales y puntos clave para obtener un esquema seguro, confiable y robusto. Otro punto importante es que existirá tanto monitoreo local, como monitoreo remoto, aprovechando los scripts sugeridos en esta tesis.

Plataforma IBM

En este esquema se busca darle más protección a todos los servidores, pero se debe tener especial cuidado con la plataforma IBM, por eso el objetivo del monitoreo está enfocado hacia estos servidores, aunque también se le instalarán herramientas de seguridad como npasswd, pgp y nullshell, a parte de los scripts de monitoreo remoto y la modificación de permisos para realizar respaldos. A estos servidores se le harán modificaciones en la distribución de aplicaciones, esto es, quedará un servidor IBM con la base de datos para el desarrollo de aplicaciones y el ambiente de pruebas y el otro servidor será utilizado solamente para la ejecución de respaldos del servidor anterior.

Servidor de desarrollo

El primer servidor se ocupará para tener una base de datos de desarrollo y por lo tanto de pruebas, esto es, aquí se harán todas las pruebas, antes de migrar a la base de datos de producción que estará en un servidor HP. También se darán de alta los usuarios necesarios para la parte de desarrollo, estos solamente se conectarán y trabajarán en el servidor en donde se alojará la base de datos, en esta parte no se requiere de un monitoreo permanente, por ser un servidor de desarrollo, a menos de que se encuentren inconsistencias o cosas anormales. En este servidor se instalarán las herramientas : npasswd, pgp y nullshell.

Servidor de respaldo

El otro servidor se utilizará para realizar los respaldos sobre el servidor de desarrollo y sólo se conectara el usuario responsable de los respaldos, el cual lo hará directamente en la consola, para que desde ahí se realicen los respaldos, y asi evitar riesgos en la comunicación. Entre los servidores IBM si habrá un monitoreo permanente, ya que se habilitarán servicios remotos para la comunicación entre ellos y con esto se evitará enviar contraseñas a través de la red. En este servidor no se instalarán herramientas de seguridad por que no es necesario, ya que solo existirá un usuario, sólo se modificarán los permisos de los dispositivos de respaldo.

Plataforma SUN

Debido a las características de la plataforma Sun (tecnología, memoria y procesador), se eligió para la parte de monitoreo y respaldos. En esta plataforma si habrá cambios más relevantes, ya que uno de los servidores se ocupará exclusivamente para realizar las siguientes tareas : monitorear el tráfico de la red, verificar probables huecos de seguridad y también servirá como repositorio de bitácoras, mientras que el otro servidor Sun se utilizará como servidor de respaldos del ambiente de producción y del servidor Web, los cuales son más críticos que el ambiente de desarrollo. Lo anterior se hará debido a que la comunicación entre los servidores Sun y HP es segura, por eso es que la plataforma IBM queda totalmente aislada de estas.

Servidor de monitoreo

El monitoreo de este servidor será muy similar al del esquema anterior, sin embargo tendrá funciones adicionales, como : almacenamiento de bitácoras y repositorio para compartir archivos entre las plataformas IBM y las de Sun y HP. En este servidor se instalarán las siguientes herramientas : saint, pgp, secure shell y obviamente el sniffer, el cual ya viene incluido en el sistema operativo del servidor Sun. A diferencia del esquema anterior, este servidor contará con más seguridad. También será el encargado de monitorear el trafico entre los servidores IBM y verificar los huecos de seguridad de todas las plataformas. En el primer caso se va a monitorear las conexiones que se establezcan entre los usuarios y el servidor (IBM) de desarrollo, además del tráfico entre el servidor de respaldo y el servidor de desarrollo, ambos también de la plataforma IBM; en el segundo caso, se va a verificar de manera constante todas las plataformas por medio de la herramienta saint, la cual realizará lo siguiente : detectar probables huecos de seguridad, recolectar información de todos los servidores y por último, generar reportes sobre la información que se vaya obteniendo

Servidor de respaldo

El servidor Sun restante será utilizado solamente para respaldar los servidores HP, los cuales tendrán la base de datos de producción y el servidor Web. Estos respaldos se podrán realizar de manera local o de manera remota, será indiferente por la seguridad que tendrá este servidor, al instalarle las herramientas sudo y secure shell. Cabe mencionar que los servidores que se ocuparán para respaldar (IBM y Sun) pueden programar cron´s para la ejecución de las tareas de respaldos, solo que en este esquema se menciona el caso de que sea necesario la intervención de un operador, por que en ocasiones se requiere mas de una cinta para completar estos.

En este servidor existirá solamente un usuario con los privilegios necesarios para ejecutar los respaldos, por lo tanto, no se le modificará permisos a ningún dispositivo y tampoco será necesario monitorear la ejecución de los respaldos, gracias a la seguridad que brindan las herramientas instaladas.

Plataforma HP

En este esquema los servidores HP tendrán distribuidas las aplicaciones más críticas, las cuales son : la base de datos de producción y el servidor Web. La decisión de utilizar estos servidores, se tomó principalmente por la seguridad que viene incluida con el sistema operativo de HP, además de las herramientas de seguridad que se le instalarán para lograr que sea mucho más robusto y por lo tanto confiable.

Este servidor contará con mucha seguridad y no será tan independiente, por que va a ser monitoreado por el servidor Sun, aunque esta parte será transparente para el servidor HP, ya que no habrá cambios en su configuración original. En esta plataforma se instalarán las siguientes herramientas : secure shell, portsentry, tcp-wrappers y tripwire.

Servidor de producción

Este servidor tendrá la base de datos de producción, esto quiere decir que es el más crítico, por lo tanto, se deberá tener mucho cuidado con la selección de herramientas de seguridad que se le instalarán. Aquí no se conectarán usuarios de desarrollo, si no usuarios que ocupen directamente las aplicaciones, por medio de una terminal segura como es secure shell. También se le van a instalar las herramientas de seguridad : tcp-wrappers, portsentry y tripwire.

Más adelante se detallará más sobre la configuración de cada herramienta de seguridad en este esquema, por que como es el servidor más delicado, se debe de cuidar su performance y su buen funcionamiento. A grandes rasgos, tcp-wrappers vigilará las conexiones hacia este servidor, portsentry evitará intentos de barridos de puertos y por último tripwire tomará una fotografía (snapshot) cada cierto tiempo de los archivos más críticos del sistema.

Servidor Web

La idea de tener un servidor Web separado de las otras aplicaciones, surgió por el gran riesgo que implica este servicio al ser uno de los más vulnerables y atacados, por eso al igual que el otro servidor HP, se seleccionarán herramientas de seguridad adecuadas y el tipo de configuración que se le asigne, evitando que quede desprotegido o vulnerable. Como las peticiones que se harán hacia este servidor, serán directamente a un puerto en especifico nada más, entonces se debe de vigilar que realmente sólo se hagan las peticiones hacia éste y en caso de no ser así, poder detectar cualquier anomalía. A este servidor se le van a instalar las herramientas de seguridad : secure shell, tcp-wrappers, portsentry y tripwire.

Cada herramienta de seguridad tendrá una función específica para este servidor Web, la cual se verá más adelante, por el momento será suficiente mencionar que tcp-wrappers limitará que sólo se puedan realizar conexiones hacia el servidor web, portsentry evitará intentos de barridos de puertos y tripwire tomará una fotografía (snapshot) cada cierto tiempo de los archivos más críticos del sistema y de los que ocupe el servidor Web.

5.3.5 Selección de un esquema para la entidad

Después de que se definieron cada uno de los esquemas de seguridad propuestos, ahora se seleccionara el más adecuado para la entidad. Es notorio que cada uno tiene sus ventajas y desventajas, sólo es necesario saber cual utilizar, dependiendo de las necesidades del lugar donde se vaya a aplicar, por eso no se puede determinar cual de los tres esquemas es el mejor, si no solamente se puede escoger el más adecuado.

Para el caso de la entidad gubernamental, el esquema más conveniente y apropiado es el tercero, tal vez es el más complejo de configurar, sin embargo la inclinación hacia éste se debe a varios factores que influyen como : tipo de servidores, las aplicaciones que tiene cada servidor, las diferentes plataformas que existen, el esquema actual de seguridad con herramientas comerciales, la administración centralizada de seguridad que debe existir y, como valor agregado, el monitoreo hacia cada uno de los servidores.

A continuación se dará el objetivo de esta tesis, que es la integración y configuración del esquema 3 en la entidad gubernamental, para lograr un ambiente de seguridad robusto y que sea mucho más confiable que el ambiente que tiene las herramientas de seguridad comerciales.

5.4 INTEGRACIÓN Y CONFIGURACIÓN DEL ESQUEMA DE SEGURIDAD UTILIZANDO HERRAMIENTAS DE LIBRE DISTRIBUCIÓN EN EL CENTRO DE CÓMPUTO DE LA ENTIDAD GUBERNAMENTAL

En esta ultima parte de la tesis, se integrarán las herramientas de seguridad en la entidad gubernamental, primero se dará una breve reseña de cada herramienta y después se configurara dependiendo de lo que se requiera en cada plataforma, en algunos casos alguna herramienta estará instalada en más de un servidor pero el tipo de configuración será diferente.

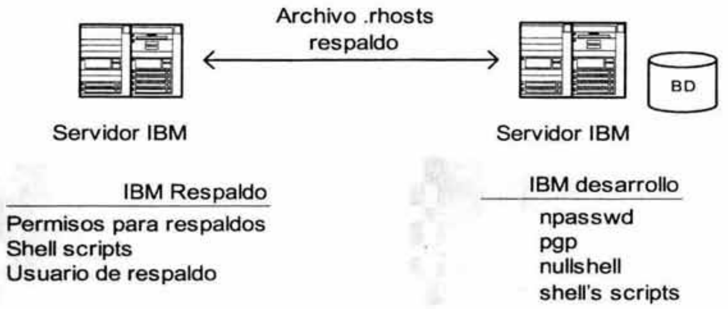


Figura 5.6 ESQUEMA DE SERVIDORES IBM

5.4.1 Plataforma IBM

En esta plataforma se utiliza un servidor de desarrollo y uno de respaldo (figura 5.6), para el primer caso se instalarán y configurarán las herramientas : npasswd, pgp y nullshell, pero en el segundo caso solo se establecerán permisos a los dispositivos para realizar los respaldos.

5.4.1.1 Servidor de desarrollo

Npasswd

Ésta es una herramienta que se puede compilar en un servidor IBM, por que no requiere de tantas librerías. Esta herramienta servirá para que en esta plataforma haya un poco de seguridad y evitar que existan contraseñas débiles, sobretodo por que es un servidor que lleva mucho tiempo en esta entidad, por lo tanto, muchos usuarios llevan ese mismo tiempo trabajando aquí, y seria muy raro que hayan cambiado su contraseña o seleccionado alguna difícil.

También se considera indispensable esta herramienta por la falta de comunicación segura, ya que la conexión se hará a través del inseguro telnet, y no con secure shell, por lo tanto, es muy viable que se intercepte la contraseña de cualquier usuario y puedan emplearla como punto de ataque malintencionado a estos servidores. Con esta herramienta se le obligará al usuario a cambiar su contraseña cada cierto tiempo y con esto se evitará que intenten conectarse con una contraseña conocida durante mucho tiempo.

Como este servidor será para el área de desarrollo, continuamente se harán peticiones de conexión por el puerto de telnet hacia el servidor, entonces es conveniente instalar esta herramienta para que cada que se cambie una contraseña, ésta sea asignada por la herramienta npasswd, y se obtenga una robusta y confiable.

La configuración de esta herramienta en este servidor quedará con los parámetros de default, excepto por estos cambios :

```
## intentos de asignación
MatchTries        5
## Ruta del diccionario (depende de cada administrador)
passwd.Dictionaries   $RUTA/dictionaries
passwd.Help           $RUTA /passwd.help
## Longitud de la contraseña
passwd.MinPassword   6
passwd.MaxPassword   10
## No permitir espacios en blanco en la contraseña
passwd.WhiteSpace     no
```

La siguiente salida es una prueba de funcionalidad de la herramienta npasswd en este servidor :

```
# /usr/lib/passwd/checkpassword

Password to check: hola123
Password bad: lower case only passwords not allowed.

Password to check: Hola123
Password bad: needs more punctuation characters or whitespace.

Password to check: H0la.123
Password ok.
```

Nullshell

Esta herramienta también se puede compilar en este servidor sin ningún problema, además de que va muy ligada con la herramienta de npasswd, puesto que servirá de ayuda cuando se cancele una cuenta, por mal uso de esta. Debido a que este servidor, no tendrá mucha seguridad y aparte se utilizará para el desarrollo de aplicaciones y de pruebas, será muy normal deshabilitar cuentas porque se detecte algo raro en su uso.

Antes de compilar la herramienta se tiene que definir la configuración en el archivo fuente (nullshell.c), en este caso quedará con la misma, solo se cambiará el mensaje que aparece al intentar conectarse, el cual quedará de la siguiente manera :

```
printf( "\n\ Cuenta desactivada, contacte al administrador de este servidor.\n\n");
```

Entonces al intentar realizar la conexión aparecerá :

```
# telnet .
Trying...
Connected to ..
Escape character is '^]'.
Local flow control on
Telnet TERMINAL-SPEED option ON

login: fabian
Password:
Please wait...checking for disk quotas

Cuenta desactivada, contacte al administrador de este servidor.
```

Pgp

Esta herramienta será de gran utilidad por que no habrá comunicaciones seguras por medio de programas como el secure shell. Para el caso de este servidor, se ocupará para depositar archivos en el servidor Sun, que servirá como repositorio de archivos firmados con la herramienta de PGP, para que de este lado solamente la pueda descifrar el destinatario. Los usuarios que necesiten estos archivos tendrán su cuenta en el servidor Sun y de ahí podrán obtener los archivos correspondientes.

El procedimiento completo se efectuará así : Primero en el servidor de desarrollo se cifrará con PGP el archivo deseado, después se va a establecer una conexión insegura (por medio de telnet) hacia el Sun, depositando el archivo cifrado con PGP. El usuario que desee tener este archivo se conectará al servidor Sun y lo descifrará también con PGP, de aquí los podrá transferir hacia los servidores de producción de manera segura por medio de secure shell, todo el flujo se muestra en la figura 5.7 :

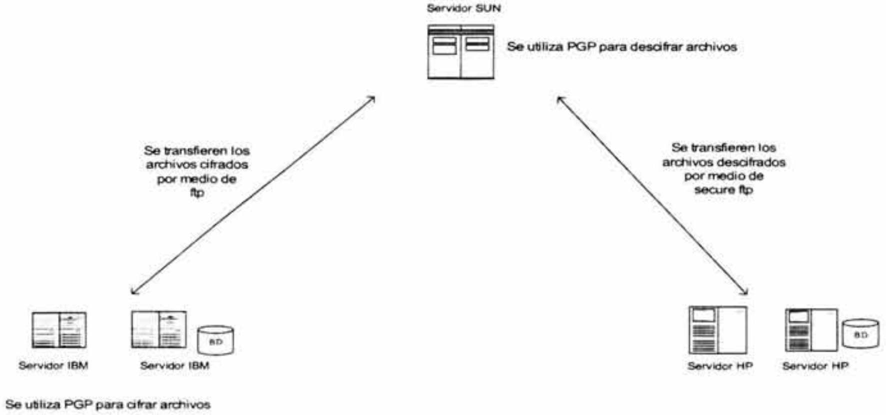


Figura 5.7 CONEXIÓN ENTRE SERVIDORES DE DESARROLLO-SERVIDOR SUN DE MONITOREO-SERVIDORES DE PRODUCCIÓN USANDO PGP Y SECURE SHELL

A continuación se hará un breve resumen de cómo se pueden cifrar archivos. Para cifrar un archivo se debe especificar el usuario destino, por medio de sus llaves públicas, para que sólo ese usuario pueda descifrarlo.

```
# pgp -e desarrollo.txt
```

Se necesita un identificador para encontrar la clave pública del destinatario, para lo que pedirá un identificador del destinatario:

```
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.

You need a pass phrase to encrypt the file.
Enter pass phrase:
Enter same pass phrase again:
```

```
Transport armor file: desarrollo.txt.asc
```

Y genera el archivo desarrollo.txt.asc, el cual estará cifrado. Este archivo ya se puede enviar al servidor Sun, aunque sea por medio de FTP, puesto que ya estará seguro e indescifrable, excepto por el destinatario.

Shell's scripts

Esto no es una herramienta de libre distribución, pero se emplearán scripts que arrojen resultados muy similares a los de las herramientas cops y tripwire, para obtener los probables huecos de seguridad que existan. Las herramientas cops y tripwire no se pueden instalar en los servidores IBM, sin embargo, esto no será una restricción para revisar a través de scripts la siguiente información : archivos con permisos comprometedores, tablas de cron, revisión de los archivos /etc/passwd y /etc/group y la búsqueda de otros archivos peligrosos.

Al igual que la herramienta cops, los scripts que aquí se mencionarán solo serán para detectar los probables huecos de seguridad, pero no corregirán absolutamente nada, esto es, solo será preventiva, mas no correctiva, debido a que en algunos casos habrá archivos que serán puestos intencionalmente por el administrador, para realizar alguna tarea especifica. La seguridad no significa tener deshabilitados todos los servicios que brinde el sistema operativo Unix o los que puedan ser riesgosos, si no de vigilar y detectar los servicios que no se deban utilizar.

Permisos en archivos

Este script revisará los permisos en todos los archivos del sistema para verificar que no existan algunos de gran riesgo como el SUID, el SGID, el sticky bit y permisos 777 principalmente cuando son del sistema. Este script al principio arrojará gran cantidad de información, pero se irá depurando conforme se vayan corrigiendo y detectando que directorios no requieren una revisión continua.

Este shell simplemente hará una búsqueda en todo el sistema, y después, poco a poco se irá afinando, según se vaya requiriendo. De preferencia la ejecución deberá ser los fines de semana o cuando haya poca actividad, debido a que consume muchos recursos de CPU y esto podría afectar en gran medida al servidor.

La realización del script dependerá de la manera de utilizar los comandos de unix de cada administrador, pero básicamente hará una búsqueda en todo el servidor de permisos peligrosos en archivos.

```
Para el SUID :
find / -perm -u+s -exec ls -la {} \;

Para el SGID :
find / -perm -2000 -exec ls -la {} \;

 Para el Sticky bit encendido :
find / -perm -o+t -exec ls -ld {} \;

Para los permisos de escritura para todo mundo:
find / -perm -o+w -exec ls -la {} \;
```

Tablas de cron (crontab)

La finalidad de este punto es realizar un script que verifique quien tiene privilegios para poder ejecutar el comando cron y además cuidar los permisos de los directorios y archivos que son utilizados. También se recomienda verificar las tablas de cron de los usuarios, para detectar que trabajos pendientes se ejecutan en el servidor.

El primer shell se encargara de revisar el archivo cron.allow, haciendo una comparación con el original para detectar si ha sido modificado, éste se recomienda ponerlo en un cron para que sea ejecutado semanalmente y revisar que coincida la lista con los usuarios que realmente necesitan usar el cron :

```
#!/usr/bin/ksh

echo "La siguiente lista de usuarios no coincide con la anterior : " > res_users_cron.txt
echo " " >> res_users_cron.txt
diff cron.allow cron.allow.comp | grep -v '[1-9]' >> res_users_cron.txt

cp cron.allow cron.allow.comp
```

El segundo script va a desplegar que crontabs tiene cada usuario, para detectar si se está haciendo mal uso de éste o si se están ejecutando programas que puedan perjudicar el performance del servidor. Se recomienda revisarlo semanalmente.

```
#!/usr/bin/ksh

cat passwd | cut -f1 -d":" > users

echo "La siguiente lista muestra los crontabs de todos los usuarios del servidor : " > trabajos
echo " " >> trabajos

for i in `cat users`
do
echo "Crontab del usuario : " $i >> trabajos
crontab -l $i  >> trabajos
echo "------------------------------------------" " >> trabajos
done

rm -rf users
```

Archivos /etc/passwd y /etc/group

Cuando se adquieren privilegios de superusuario en un servidor, los primeros archivos que se llegan a modificar son los siguientes : /etc/passwd y /etc/group, por eso es conveniente revisarlos de manera constante para detectar si han sufrido cambios. Estos se deben verificar por medio de un shell script y ocupando utilerías de unix : la primera revisión que se hará es sobre los permisos, los cuales deberán de tener 644 o rw-r—r—, esta revisión será semanal,

```
#!/usr/bin/ksh

echo "Permisos de los archivos /etc/passwd y /etc/group : " > res_perm_archs.txt
echo " " >> res_perm_archs.txt

ll /etc/passwd /etc/group | tr -s "  " " " |cut -f1,9 -d" " >> res_perm_archs.txt
```

Después se verificarán los campos más importantes, que para ambos son los primeros 3, que a continuación se detallan :

Nombre de usuario : Una de las primeras acciones que realiza un hacker, es dar de alta a un usuario en el sistema, para que las próximas conexiones al servidor las haga a través de éste. De este campo se debe revisar que no exista algún usuario adicional que no fuera dado de alta por el superusuario; cuando no se lleva un control de estos cambios, se deberá sacar una diferencia entre el archivo actual y una copia de éste mismo con una semana de antigüedad (o menos), y así sucesivamente, para saber que usuarios han sido añadidos o quitados. Al igual se recomienda ejecutar este script semanalmente.

```
#!/usr/bin/ksh

cat /etc/passwd | cut -f1 -d":" > logins

echo "Usuarios del archivo /etc/passwd que han cambiado : "        > res_logins.txt
echo "La siguiente lista de usuarios no coincide con la anterior : " >> res_logins.txt
echo " "                                   >> res_logins.txt

diff logins logins.comp | grep -v '[1-9]' >> res_logins.txt

cp logins logins.comp
```

Contraseña : Éste es un campo que también suele modificarse, aunque aquí la contraseña se presenta de manera cifrada, es muy común que se borre el contenido de este campo, esto quiere decir, no se puede cambiar la contraseña desde aquí, si no solamente anularla, para que ésta ya no exista. Entonces, al usuario que se le borre este campo, podrá entrar al servidor sólo con su login, por que ya no tendrá contraseña.

La revisión que se hará sobre este campo, será solamente que exista, no se debe realizar una comparación con otro archivo, ya que es normal que este campo se modifique muy seguido, debido a que los usuarios pueden cambiar su contraseña las veces que ellos lo deseen. La revisión se hará semanalmente.

```
#!/usr/bin/ksh

echo "Validación de la existencia de las contrasen~as de los usuarios: " > res_contra_users.txt
echo " " >> res_contra_users.txt

cat /etc/passwd| cut -f1,2 -d":" >> res_contra_users.txt
```

UID ó GID : Otro de los campos que tienen que revisarse es del UID o GID. Aquí se debe verificar que ninguna cuenta, excepto la del super-usuario (root) deberá tener el UID = 0 ó el GID = 0, por que esto significa que tiene privilegios de root o que pertenece al grupo de root. A pesar de ser un problema crítico se recomienda revisarlo semanalmente.

```
#!/usr/bin/ksh

echo "Validacion de la existencia de UID = 0 o GID = 0 " > res_uid.txt
echo " " >> res_uid.txt

echo "En el archivo /etc/passwd : " >> res_uid.txt
echo " " >> res_uid.txt
cat /etc/passwd | cut -f1,3 -d":" | grep ":0" >> res_uid.txt

echo " " >> res_uid.txt
echo "En el archivo /etc/group : " >> res_uid.txt
echo " " >> res_uid.txt
cat /etc/group | cut -f1,3 -d":" | grep ":0" >> res_uid.txt
```

Otros archivos

Otros archivos que pueden ser grandes huecos de seguridad, son el /etc/hosts.equiv y el .rhosts, como se había mencionado en capítulos anteriores. El script para la revisión de estos archivos es muy sencillo, por que solo se debe revisar si existen, si es así, se deberán de borrar inmediatamente o renombrar e investigar cuando, por que y quien los puso en el sistema. Para este servidor en especial, si van a existir archivos de este tipo, pero sólo para un usuario en especial, que será encargado de los respaldos (el usuario backup), entonces la revisión se hará sobre los demás usuarios y semanalmente.

```
Para el .rhosts   :
find / -name ".rhosts" -exec ls -la {} \;

Para el hosts.equiv
find / -name "hosts.equiv" -exec ls -la {} \;
```

Monitoreo local

Algunos comandos de monitoreo sólo se pueden utilizar de manera local y están enfocados a monitorear el performance y uso del servidor, aquí sólo se revisará el espacio en disco :

Espacio en disco :

Es sumamente importante en este servidor monitorear el espacio en los filesystems, por tratarse de un servidor de desarrollo de aplicaciones y de pruebas. Aunque parezca algo muy básico y muy trivial, tener un monitoreo constante al espacio almacenado, permite prevenir muchos problemas e inclusive evita que se pueda caer el servidor en caso de llenarse el filesystem de root.

df –k : Por medio de este comando, se puede realizar un script para que envié una alarma al momento de que alguno de los filesystems críticos sobrepase cierto umbral. Tal vez será necesario monitorear todos los filesystems con umbral crítico mayor a 85% de uso, de cumplirse esta condición se envía inmediatamente una alarma, como se muestra en el siguiente script :

```
#!/usr/bin/ksh

df -k | grep -v used | tr -s "      " " " | cut -f5,6  -d" " | tr -s "%" " " > cuatro

awk '$1 > 85 {print $1 $2}' cuatro > resulta.txt

VAR=`ll resulta.txt | tr -s "   " " " | cut -f5 -d" "`

if [ VAR -eq 0 ]
then
exit
else
write monitor <<EOT
Algunos filesystems sobrepasan el 85% de su uso, verificarlo en el archivo : resulta.txt
EOT
fi
```

Todos los shell scripts arrojan los resultados hacia archivos temporales para su posterior revisión, más adelante en revisión de bitácoras se propone automatizar esta verificación, por medio de comandos de unix o si es muy crítico el problema, que se notifique de manera inmediata por medio de una alarma hacia alguna terminal.

Monitoreo remoto

El monitoreo remoto se hará de la plataforma Sun hacia esta plataforma, por lo tanto, este tema será detallado en la parte de esta plataforma.

5.4.1.2 Servidor de respaldo

Permisos en dispositivos y comandos.

Con la herramienta sudo, es fácil restringir el uso de comandos a un determinado grupo de usuarios, pero en este servidor no será instalada esta herramienta, sin embargo, esto no significa que no se pueda limitar el uso de comandos y dispositivos a los usuarios normales.

Como se mencionó anteriormente, los respaldos son obligatorios en cualquier servidor, en este caso no es la excepción, pero no serán ejecutados por el superusuario, si no por otro usuario (con username backup), para esto es necesario restringir los permisos en dispositivos y comandos.

El usuario encargado de realizar los respaldos se le deberá poner el mismo username en ambos servidores (backup), tanto en este servidor como en el servidor de desarrollo, para que pueda ocupar los servicios de red remotos rlogin, rsh y rcp, además de habilitarle el acceso en los archivos de configuración (.rhosts, hosts.equiv y /etc/hosts).

Los archivos que se van a respaldar se pueden traer desde el servidor de desarrollo al servidor de respaldo a través de un remote copy o por medio de ftp, y entonces, se procede a ejecutar el respaldo.

Para poder realizar los respaldos de manera segura es necesario modificar los permisos en los dispositivos (/dev/rmt/#m), para que estos solo se puedan ser usados por cierto grupo de usuarios.

En este servidor los dispositivos se llaman de la siguiente manera :

/dev/rmt

crw-rw-rw- tm0n
crw-rw-rw- tm0

Deberán de quedar :

/dev/rmt

crw-rw---- tm0n
crw-rw---- tm0

De esta manera se podrá restringir el uso de las unidades de cinta, pero para tener más seguridad es necesario ponerle los mismos permisos a los comandos que se vayan a usar para respaldar, excepto al comando tar, por que es un comando que se usa muy a menudo y no necesariamente escribe a cinta.

Los comandos a los que se les cambiarán los permisos (550), para que solo puedan ser ejecutados por el usuario root y por el grupo al que pertenezca el usuario backup, quedarán así :

dump	**-r-xr-x---**
restore	**-r-xr-x---**
ufsdump	**-r-xr-x---**
cpio	**-r-xr-x---**

Shell's scripts

Al igual que en el servidor anterior, en éste también se utilizarán shell's scripts para detectar problemas de seguridad, aunque no se ocuparán los mismos debido a que no es un servidor crítico y solamente se ocupará para realizar respaldos. Los scripts que se utilizarán son los siguientes :

- El script que valida la existencia de las contraseñas
- El script que verifica la existencia de UID=0 o GID=0
- Búsqueda de archivos .rhosts y hosts.equiv (excepto para el usuario backup)

Su revisión será también semanal y la información obtenida de estos scripts, también se almacenarán en archivos temporales para su posterior revisión, la cual también se automatizara más adelante.

Monitoreo remoto

Para este servidor de respaldo el monitoreo remoto también se realizará desde la plataforma Sun, por lo tanto, este tema será detallado más adelante.

5.4.2 Plataforma Sun

Esta plataforma es la que llevará el control de los respaldos de producción y también el monitoreo hacia los demás servidores, por lo tanto, es muy importante la configuración de este servidor. Al servidor de respaldo solamente se le instalarán 2 herramientas : secure shell y sudo, mientras que el otro servidor tendrá las siguientes : secure shell, pgp, saint y el sniffer propio de Sun. El servidor de monitoreo no solo servirá como tal, si no tendrá una función secundaria, la de actuar como intermediario en transferencia de archivos importantes (repositorio), el esquema de esta plataforma se muestra en la Figura 5.8.

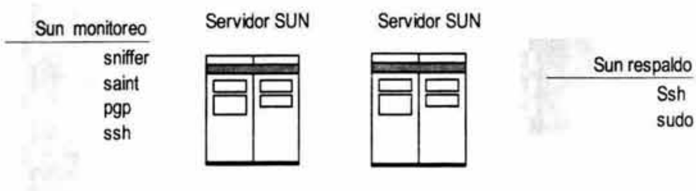


Figura 5.8 ESQUEMA DE SERVIDORES SUN

5.4.2.1 Servidor de respaldo

Secure shell

Esta herramienta es considerada la mas importante por las funciones que brinda y por que será la mas utilizada por los usuarios, que hasta el momento se conectan a los servidores por medio del telnet. Este no es un cambio tan drástico para los usuarios, por que presenta un emulador muy parecido al telnet de windows. Esta herramienta esta pensada en los usuarios que trabajan con pc's y sistema operativo Windows. El cambio hacia esta herramienta será muy satisfactorio para los usuarios, por que inmediatamente notarán la diferencia, sobretodo por el ftp seguro que trae incluido, el cual es totalmente gráfico.

En este servidor el secure shell se utilizará para poder conectarse de manera remota y poder ejecutar los respaldos sin estar físicamente en el centro de computo, solamente cuando haya que cambiar las cintas.

El archivo de configuración del secure shell, para este servidor quedara con los parámetros de default, solo cambiarán los siguientes parámetros :

VerboseMode ***yes***

Este opción sirve para imprimir mensajes de como se establece la conexión con el servidor . Puede ser de gran ayuda para detectar problemas. Es conveniente tener esta opción siempre activada, para ver todo el procedimiento de conexión.

Port ***40***

Aquí se le indica que puerto de conexión se va a ocupar para secure shell, por default es el 22, sin embargo, se recomienda poner otro puerto, para este caso se ocupará el 40. Del lado del cliente se le debe indicar que se conectará por este puerto.

KeepAlive

Con esta opción activa, el servidor evita que se queden sesiones colgadas, previa notificación al cliente. Como éste es un servidor de respaldo, no es necesario activar esta opción.

MaxConnections

Aquí se especifica el número máximo de conexiones que secure shell establece al mismo tiempo. Esta opción es muy útil cuando se desea evitar un ataque de muchas conexiones al servidor y que originen que se comporte inestable por los recursos que se estén agarrando o inclusive que logre tirar al servidor. Ponerle el valor de "0" significa que tendrá conexiones ilimitadas. Para este servidor en particular se dejarán solo 10 conexiones, por que sólo se usará para le ejecución de respaldos, por lo tanto, no requiere de muchas sesiones.

Ciphers

Indica que cifrado se utilizará en la conexión, son soportados : aes, blowfish, twofish, arcfour, cast, 3des, y des. Aquí se dejará AnyCipher, la cual puede tomar cualquiera de las anteriores.

LoginGraceTime

Con esta opción se le determina el tiempo que tarda el servidor para desconectar una sesión si ésta no fue exitosa, si el valor es "0", quiere decir que no hay limite de tiempo. Para este servidor se le dejará el valor por default (600 segundos).

PermitEmptyPasswords

Aquí se especifica si se permiten cuentas que no tengan contraseñas. Aquí se recomienda siempre dejarlo en NO, para evitar que existan contraseñas vacias.

PasswordGuesses

En esta opción se le índica el número de intentos que el usuario tiene para conectarse. También se le quedará el valor que tiene por default (3).

AllowHosts

Este componente es muy útil para la seguridad del servidor, puesto que solo va a permitir que se conecten los equipos que estén listados delante de esta opción. Como a este servidor se conectan muy pocos usuarios, si deben listarse los equipos que podrán tener acceso.

AllowUsers

Igual que el anterior, sólo que aquí se van a listar los usuarios que se les permitirá la conexión. También se listarán los usuarios que podrán conectarse para ejecutar los respaldos, en este caso el usuario backup deberá estar listado aquí.

AllowGroups

Esta opción es idéntica a las 2 anteriores, sólo que aquí se maneja el permiso hacia todo un grupo. Esta se dejará vacía, por que son pocos usuarios.

PermitRootLogin ***yes***

Indica si se podrá o no establecer una conexión con el usuario root desde el cliente. Esta opción siempre debe ir, ya que se toma como medida de seguridad en cualquier servidor que root no se pueda conectar directamente a través de una consola.

ChRootUsers ***ftp,guest***

Aquí se especifica una lista de usuarios a los que se les dará un ambiente de chroot. Lo anterior significa que un usuario solo podrá moverse desde su directorio $HOME hacia abajo. Esta opción no se activará ya que no es un servidor crítico.

Nota : Se recomienda proteger al servidor por medio de secure shell, pero sin tomarlo como reemplazo a las otras herramientas de seguridad o a las del propio sistema operativo, ya que al tirar el demonio de secure shell, no se toma en cuenta nada de lo descrito en la configuración anterior.

Sudo

La herramienta de sudo se utilizará en este servidor para que usuarios sin privilegios de root, puedan utilizar comandos de respaldo de los cuales el superusuario es dueño. Este servidor será el encargado de respaldar los dos servidores de producción : el de base de datos y el de Web, por eso es necesario configurar esta herramienta y permitir la ejecución segura de respaldos críticos.

Como éste es un servidor de respaldos, entonces el archivo de configuración (sudoers) quedará de tal forma que sólo un usuario pueda usar comandos para respaldar, a los cuales se les cambiará el permiso para que solo root los pueda ejecutar. El nombre del usuario será **backup**, tanto en este servidor como en los servidores de producción, con el fin de que pueda realizar una conexión desde este servidor y así poder ejecutar respaldos de manera remota o de manera local, por medio de secure shell y el ftp seguro, para posteriormente pasarlos a unidades de cinta.

Los comandos que se podrán usar para ejecutar los respaldos, son los siguientes :

```
/usr/sbin/ufsdump        -r-x------
/usr/bin/cpio            -r-x------
```

los cuales cuentan con permisos 500 para que solo root pueda ejecutarlos.

A continuación se muestra el contenido del archivo de configuración, en donde indica que el usuario backup va a poder ejecutar los comandos anteriores, a pesar de sus permisos.

```
sudoers file.

# This file MUST be edited with the 'visudo' command as root.
# See the sudoers man page for the details on how to write a sudoers file.
# Host alias specification
# User alias specification
# Cmnd alias specification
# Defaults specification
# User privilege specification
root   ALL=(ALL) ALL

# Uncomment to allow people in group wheel to run all commands
# %wheel      ALL=(ALL)     ALL
# Same thing without a password
# %wheel      ALL=(ALL)     NOPASSWD: ALL
# Samples
# %users  ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users  localhost=/sbin/shutdown -h now

backup ALL=/usr/sbin/ufsdump, /usr/bin/cpio
```

5.4.2.2 Servidor de monitoreo

Pgp

La función principal de este servidor será la del monitoreo hacia los servidores inseguros, pero además servirá de repositorio para la transferencia segura de archivos entre las plataformas IBM y HP. Los usuarios que requieran la transferencia de archivos entre plataformas, deberán cifrar la información usando PGP desde el servidor de desarrollo de IBM, después depositarán estos archivos en este servidor Sun, luego en este servidor se hará el procedimiento de descifrado (teniendo la llave publica de la persona que le está enviando este archivo) y se podrán transferir hacia la plataforma HP por medio de secure shell. El procedimiento también puede ser usado de manera inversa.

La manera de descifrar archivos es la siguiente :

 # pgp –p desarrollo.txt.asc

```
Pretty Good Privacy(tm) Version 6.5.8
(c) 1999 Network Associates Inc.
Uses the RSAREF(tm) Toolkit, which is copyright RSA Data Security, Inc.
Export of this software may be restricted by the U.S. government.

You need a pass phrase to encrypt the file.
Enter pass phrase:
Enter same pass phrase again:
```

Aquí se teclea la frase que se le dio al momento de generar las llaves :

```
Transport armor file: desarrollo.txt.asc
```

Y genera el archivo desarrollo.txt.asc, el cual estará cifrado. Este archivo ya se puede enviar al servidor Sun, aunque sea por medio de FTP, puesto que ya estará seguro e indescifrable, excepto por el destinatario.

Secure shell

En este servidor también se usará esta herramienta, para que los usuarios de la plataforma HP se conecten de manera segura y puedan intercambiar información con la plataforma IBM. Otra funcionalidad del secure shell en este servidor, será la de permitir realizar un monitoreo remoto, esto quiere decir que desde las computadoras personales, se pueda realizar el monitoreo a pesar de que haya herramientas gráficas, como es el caso de Saint, que requiere de un browser para su monitoreo. Lo anterior es otra muestra de las grandes ventajas que tiene la herramienta de secure shell, al permitir levantar aplicaciones gráficas desde una Pc y lo más importante que sea de manera segura, por medio de un "tunneling" y usando el puerto de secure shell (figura 5.9)

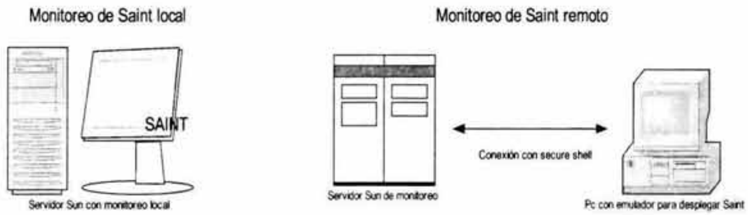


Figura 5.9 MONITOREO DESDE EL SERVIDOR (DE MANERA LOCAL) Y DESDE UNA PC USANDO SECURE SHELL Y SU TUNNELING (SAINT)

Esta herramienta también se ocupará para transferir de manera segura las bitácoras de los demás servidores hacia éste. Lo anterior se detallará al final de este capítulo.

Los parámetros que se van a modificar para este servidor son los siguientes :

VerboseMode	**yes**
Port	**50**
MaxConnections	**20**
Ciphers	**AnyCipher**
PermitEmptyPasswords	**No**
AllowHosts	**Lista de pc's**
AllowUsers	**Lista de usuarios**
PermitRootLogin	**yes**

Saint

La principal característica de este servidor, es la de realizar un monitoreo de huecos de seguridad en los distintos servidores, por este motivo la herramienta de saint se instalará en este servidor Sun de monitoreo y se ejecutará de manera constante para la detección oportuna de los huecos que puedan poner en riesgo el área de producción. Como elemento adicional, esta herramienta permite recolectar un inventario de hardware y software de los servidores a monitorear, lo cual es de mucha ayuda cuando se solicitan reportes con características de los servidores.

A continuación se muestra la configuración que tendrá la herramienta saint para monitorear a los demás servidores. Se configurará de tal manera que haga una revisión completa, con la finalidad de determinar que huecos de seguridad presenta cada uno,. Cabe señalar que en los servidores IBM se tienen habilitados algunos servicios de red para la comunicación remota, los cuales los marcara como huecos de seguridad, sin embargo no son tales, por eso se debe de realizar una revisión muy cuidadosa de los resultados para valorar que es realmente un hueco y cual no lo es. También se determinará que conviene monitorear en cada servidor. Esta revisión se recomienda hacerla mensualmente y guardar los reportes generados, para tenerlos como históricos.

Entonces, se mencionará el tipo de configuración que se dejará por default, para este esquema :

La primera opción del menú (data management) (figura 5.10) se refiere a los servidores que tiene en su base de datos la herramienta de saint, como se muestra en la siguiente imagen son los servidores mostrados en el diagrama de los servidores. En esta lista también aparece el servidor local (monitoreo_SUN), para que también sea revisado.

Figura 5.10

En la segunda opción del menú (target selection) (figura 5.11) se debe poner que equipos se van a revisar, puede hacerse de manera individual, a un rango o tomar los equipos que se encuentran en un archivo llamado target_file. Para esta configuración se dejará la opción de que verifique los archivos que están dentro del archivo target_file, posteriormente se puede cambiar para que solo revise los equipos deseados. Este último caso se recomienda después de una primera revisión general, para configurar lo que realmente es deseado monitorear en cada servidor, dependiendo del uso que tenga éste.

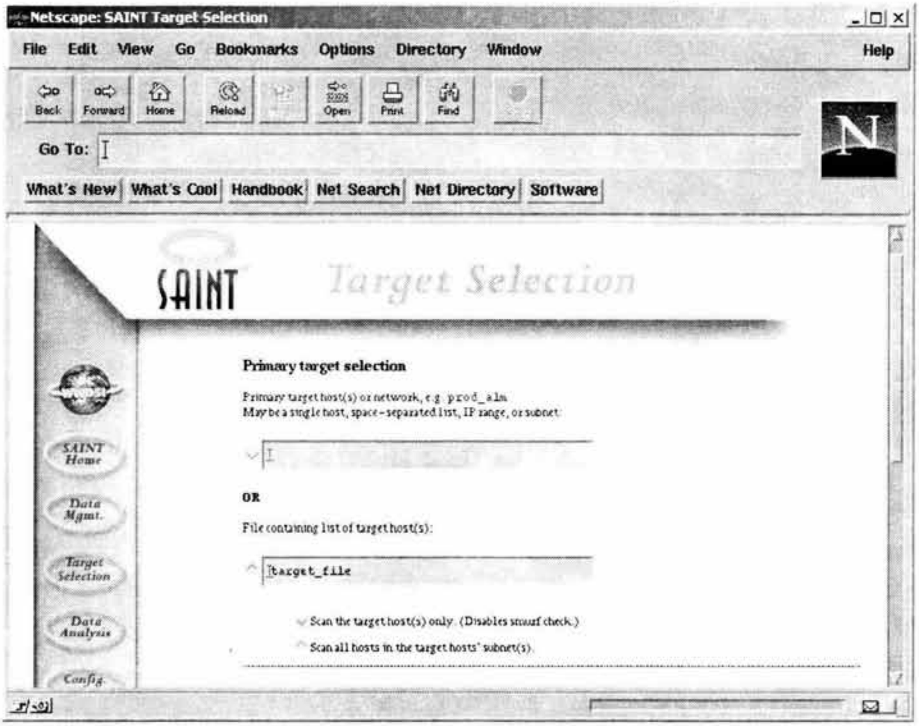


Figura 5.11

En la siguiente pantalla (figura 5.12) se muestra el tipo de revisión que se hará, este varía de acuerdo a que tan profunda se quiera realizar, puede ser desde una ligera hasta una personalizada, para este caso se usará una revisión completa (heavy +). También se le puede especificar si pasa a través de un firewall o no, en este caso no pasa por ninguno.

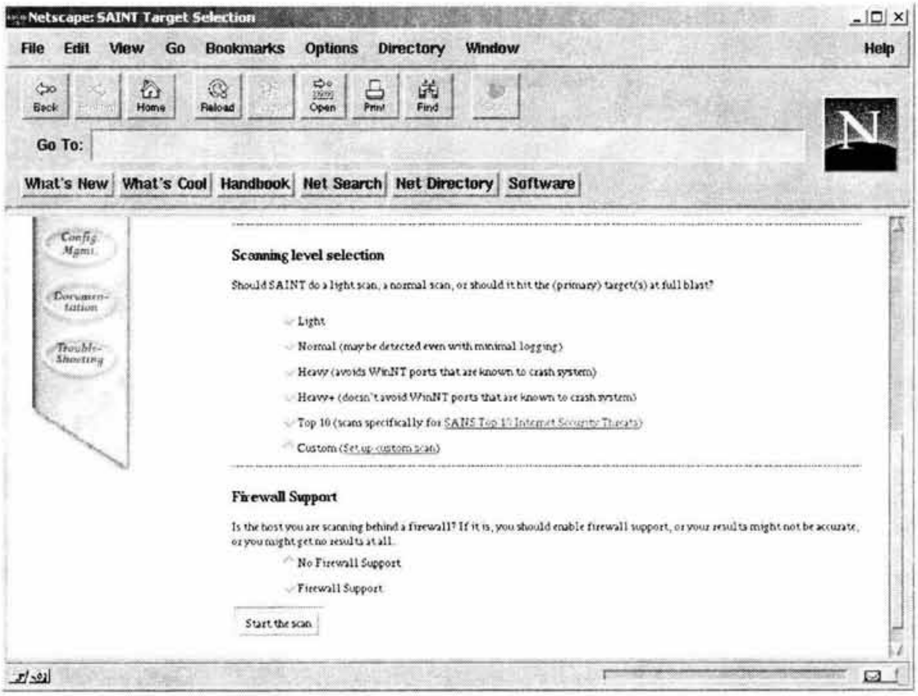


Figura 5.12

El siguiente punto (configuration management) (figura 5.13) se refiere al nivel de revisión que deberá aplicarse y los timeouts que se vana a manejar. Esta parte es muy útil, ya que se evitan problemas de comunicación con el servidor al momento de enviar paquetes de verificación.

En la pantalla siguiente se muestran los siguientes puntos :

- Se pide primero un directorio en donde se almacenarán los resultados que se obtengan.
- El otro punto es el tipo de revisión que se hará por default, en esta parte se puede dejar la personalizada, para que al momento de verificar cualquier servidor se pueda seleccionar lo que nos interesa para cada uno, como se verá más adelante.
- Después pide cuantos intentos de contraseñas deberá intentar adivinar, esto es necesario por que en servidores como HP, a los 3 intentos bloquea las cuentas, y esto acarrea problemas cuando se bloquean muchas cuentas y después el administrador tiene que desbloquear todas. Para evitar lo anterior, es mejor dejarla con 2 para los servidores Sun y HP y con 5 para los servidores IBM.
- El ultimo punto de esta pantalla indica el tiempo que se tomará para el timeout, debido a que esta herramienta se manejará en una intranet y sin ningún firewall en medio, entonces se escogerá el valor fast (10), que indica que no debe tardar más de 10 segundos en poder comunicarse con el servidor a monitorear.

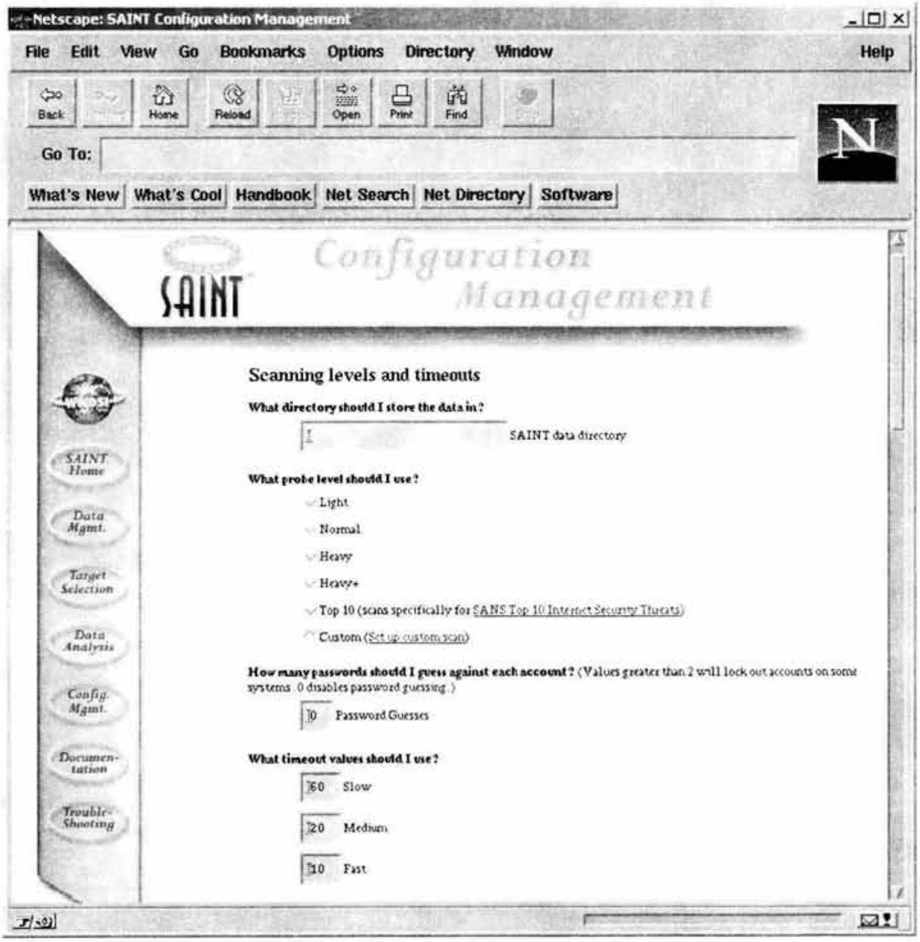


Figura 5.13

La siguiente pantalla (figura 5.14) muestra más opciones de la parte de configuration management.

- La primera se refiere a el timeout que tomará en cuenta en cada chequeo que realice Saint hacia los servidores, se dejará el mismo que el anterior (fase) por tratarse de una intranet.
- El siguiente punto se refiere al timeout para el "scanneo" de puertos TCP y UDP, se quedarán 45 segundos.
- En esta opción se le indica cuantos chequeos concurrentes puede realizar, entre mayor sea este número, requiere de más uso de memoria, en este caso como se trata de un servidor dedicado se puede poner un número elevado (10).
- Después indica que señal se le deberá enviar a un proceso para que lo termine si este llega al timeout estimado. Se le dejará el 9, para que se vea forzado a terminarlo.

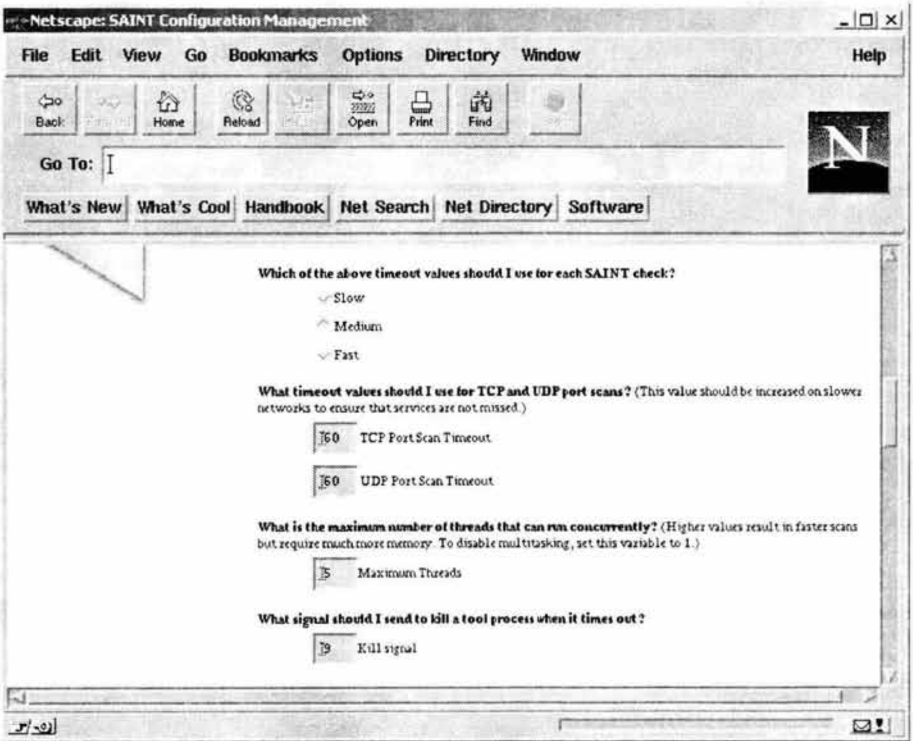


Figura 5.14

En la parte de análisis de datos (data analysis) (figura 5.15), se ven los reportes generados después de la verificación que ejecuta la herramienta. Estos reportes se generan por vulnerabilidades, por servidor o por servidores.

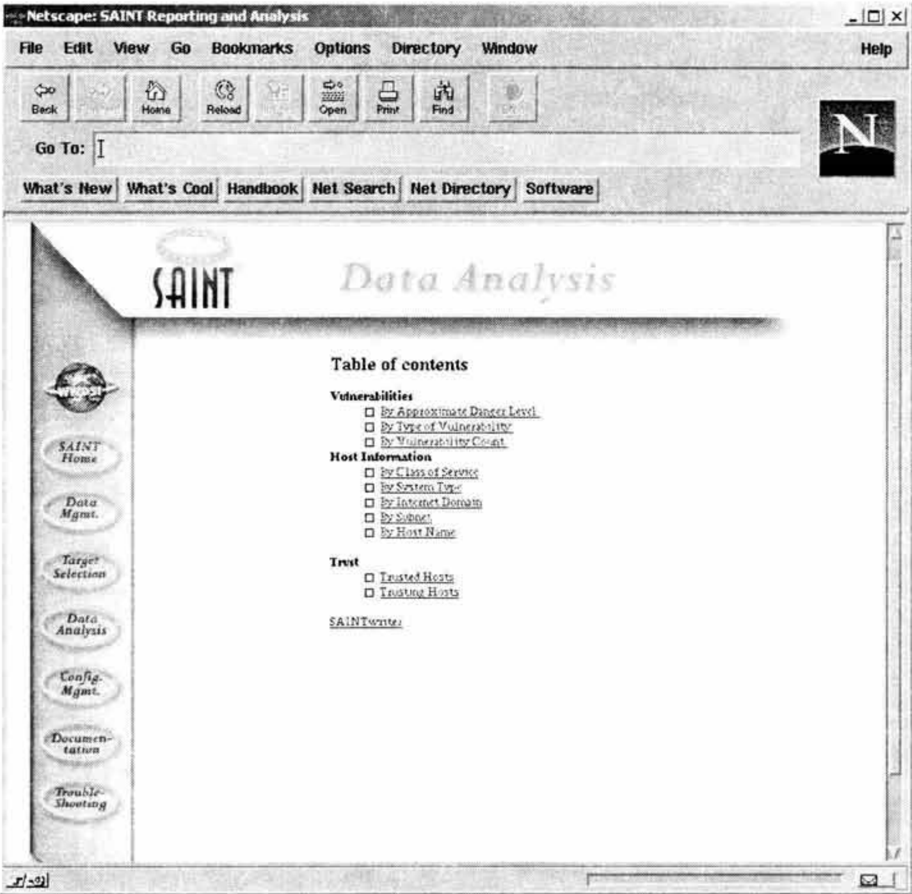


Figura 5.15

Por último, se detallará un poco acerca del archivo de configuración, el cual contiene lo que se le revisará a todos los servidores por default. Estas pantallas se verán para ver que opciones debe usarse en cada servidor (respaldo, desarrollo, web y producción) cuando la revisión sea individual (esto es cuando se escoja la configuración personalizada). A pesar de que éste es un archivo de configuración, son las mismas opciones que tiene al seleccionar la personalizada.

En la primer pantalla (figura 5.16.1) aparece :

- Como primera opción aparece el tipo de verificación que se realizará por default, se pondrá la misma que en el target selection, la que hace una revisión completa (heavy +).
- Después muestra una serie de servicios de red que ofrece el servidor (figuras 5.16.2 y 5.16.3), de los cuales quedaría de la siguiente manera :

server/ servicio	desarrollo_ibm	resplado_ibm	monitoreo_sun	respaldo_sun	web_hp	producción_hp
rpc	*	*	*	*	*	*
ddos	--	--	--	--	--	--
dns	--	--	--	--	*	--
finger	*	*	*	*	*	*
ostype	*	*	*	*	*	*
tcpscan	*	*	*	*	*	*
udpscan	*	*	*	*	*	*

*** A estos servidores si se les revisará**

En los recuadros de los puertos TCP y UDP, irán todos los que estén dentro del archivo /etc/inetd.conf de cada servidor.


Netscape: Custom Scan Configuration
File Edit View Go Bookmarks Options Directory Window Help
Back Home Reload Open Print Find
Go To:
What's New What's Cool Handbook Net Search Net Directory Software
SAINT
Configuration Management
Reset custom configuration to default level.
light
Reset
Core probes
The selected probes run against all targets. Disabling some of these probes may cause conditional probes not to run.
rpc
ddos
dns
finger
ostype
tcpscan
TCP Ports
udpscan
UDP Ports
SAINT Home
Data Mgmt.
Target Selection
Data Analysis
Config. Mgmt.
Documentation
Trouble-Shooting

Figura 5.16.1

A continuación se muestran una serie de servicios que también se revisarán basados en los resultados del punto anterior.

server/ servicio	desarrollo_ibm	resplado_ibm	monitoreo_sun	respaldo_sun	web_hp	producción_hp
backdoors	*	*	*	*	*	*
boot	*	*	*	*	*	*
cim	--	--	--	--	--	--
cmsd	--	--	--	--	--	--
dns_chk	--	--	--	--	*	--
ftp	*	*	*	--	--	--
ftp_bounce	*	*	*	--	--	--
gopher	--	--	--	--	--	--
http	--	--	--	--	*	--
imap	--	--	--	--	--	--
inn	--	--	--	--	--	--
irix	--	--	--	--	--	--
login	*	*	*	*	*	*
mountd	*	*	*	*	*	*
nfs_chk	*	*	*	*	*	*
pop3	*	*	*	*	*	*
printer	*	*	*	*	*	*
relay	*	*	*	*	*	*
rex	--	--	--	--	--	--
rexec	*	*	*	--	--	--
rlogin	*	*	*	--	--	--
rsh	*	*	*	--	--	--
rstatd	*	*	*	--	--	--
rusers	*	*	*	--	--	--
sadmind	*	*	*	*	*	*
sendmail	--	--	--	--	--	--
showmount	*	*	*	*	*	*
smb	--	--	--	--	--	--
smurf	*	*	*	*	*	*
snmp	--	--	*	*	*	*
ssh	--	--	*	*	*	*
statd	*	*	*	*	*	*
tftp	*	*	*	--	--	--
tooltalk	*	*	*	*	*	*
xhost	--	--	*	*	*	*
yp-chk	*	*	*	*	*	*
ypbind	*	*	*	*	*	*

*** A estos servidores si se les revisará**

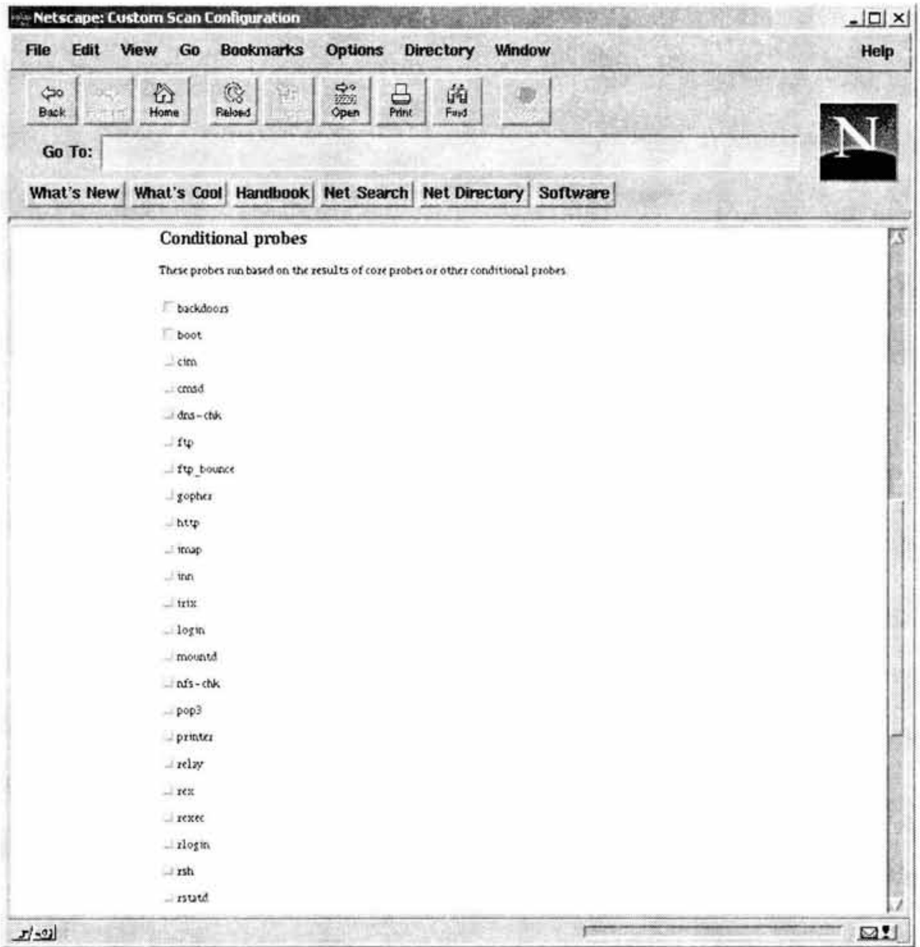

Netscape: Custom Scan Configuration
File Edit View Go Bookmarks Options Directory Window Help
Back Home Reload Open Print Find
Go To:
What's New What's Cool Handbook Net Search Net Directory Software
Conditional probes
These probes run based on the results of core probes or other conditional probes.
backdoors
boot
cim
cmsd
dns-chk
ftp
ftp_bounce
gopher
http
imap
inn
irix
login
mountd
nfs-chk
pop3
printer
relay
rex
rexec
rlogin
rsh
rstatd

Figura 5.16.2

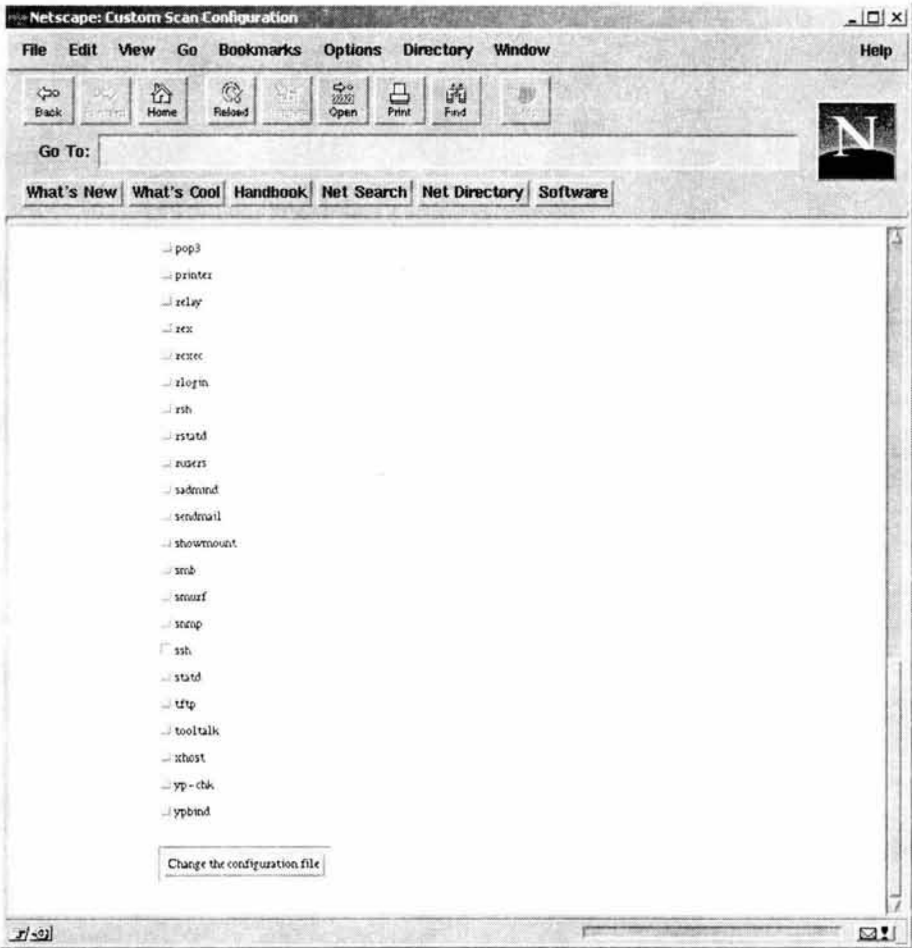

Netscape: Custom Scan Configuration
File Edit View Go Bookmarks Options Directory Window Help
Back
Home
Reload
Open
Print
Find
Go To:
What's New
What's Cool
Handbook
Net Search
Net Directory
Software
pop3
printer
relay
rex
rexec
rlogin
rsh
rstatd
rusers
sadmind
sendmail
showmount
smb
smurf
snmp
ssh
statd
tftp
tooltalk
xhost
yp-chk
ypbind
Change the configuration file

Figura 5.16.3

Sniffer

Esta herramienta adicional será de mucha utilidad, por que "vigilará" principalmente a la plataforma IBM, la cual en este esquema es la más vulnerable por carecer de herramientas de seguridad. El sniffer de Sun monitoreará constantemente a los servidores IBM y si es necesario, también se encargará de vigilar algo específico de otras plataformas.

Para el caso de la plataforma IBM, estará monitoreando las conexiones entre servidores de este tipo, por ser la parte más comprometedora al viajar en claro las contraseñas de los usuarios que se conecten a cualquiera de los dos servidores y obviamente para vigilar que en la ejecución de los respaldos no se presente algún percance.

Como se ha mencionado, se cuenta con 2 servidores IBM, uno se usará para respaldar (respaldo_IBM) y el otro para desarrollar aplicaciones (desarrollo_IBM). Lo que se deberá monitorear entre estos servidores será la comunicación, esto es, las peticiones que se hagan entre ellos, pero sólo las que puedan poner en riesgo a cualquiera de los 2 servidores, que a pesar de no ser críticos, no es conveniente dejarlos aislados y sin monitoreo.

La comunicación entre ambos servidores sólo será por los siguientes puertos :

Entonces por cada puerto usado se va a obtener un archivo de salida, a éste se le tiene que dar una revisión constante para detectar alguna inconsistencia o si existen intentos de conexión irregulares.

 # snoop -o salida.txt respaldo_IBM desarrollo_IBM

La salida que va a presentar el comando anterior, será similar a esto :

```
root# snoop respaldo_IBM desarrollo_IBM
Using device /dev/hme (promiscuous mode)
    respaldo_IBM -> desarrollo_IBM     TELNET C port=35985
    desarrollo_IBM -> respaldo_IBM     TELNET R port=35985
    desarrollo_IBM -> respaldo_IBM     FTP R port=35986 220 desarrollo_IBM FTP serve
    respaldo_IBM -> desarrollo_IBM     FTP C port=35986
    respaldo_IBM -> desarrollo_IBM     FTP C port=35986 USER fabian\r\n
    desarrollo_IBM -> respaldo_IBM     FTP R port=35986
    desarrollo_IBM -> respaldo_IBM     FTP R port=35986 331 Password require
    respaldo_IBM -> desarrollo_IBM     FTP C port=35986
    respaldo_IBM -> desarrollo_IBM     FTP C port=35986 PASS fabo321\r\n
    desarrollo_IBM -> respaldo_IBM     FTP R port=35986 530 Login incorrect.
    respaldo_IBM -> desarrollo_IBM     FTP C port=35986
    respaldo_IBM -> desarrollo_IBM     FTP C port=35986 QUIT\r\n
    desarrollo_IBM -> respaldo_IBM     FTP R port=35986 221 Goodbye.\r\n
    respaldo_IBM -> desarrollo_IBM     TCP D=22 S=35987     Ack=917549596 Seq=3100350316 Len=0 Win=8760
    respaldo_IBM -> desarrollo_IBM     TCP D=22 S=35987     Ack=917550539 Seq=3100350994 Len=0 Win=8760
    desarrollo_IBM -> respaldo_IBM     TCP D=35987 S=22     Ack=3100351266 Seq=917550539 Len=0 Win=8760
    respaldo_IBM -> desarrollo_IBM     TCP D=22 S=35987     Ack=917551131 Seq=3100351266 Len=0 Win=8760
    respaldo_IBM -> desarrollo_IBM     TCP D=22 S=35987     Ack=917551131 Seq=3100351266 Len=16 Win=8760
    desarrollo_IBM -> respaldo_IBM     TCP D=35987 S=22     Ack=3100351282 Seq=917551131 Len=0 Win=8760
    respaldo_IBM -> desarrollo_IBM     TCP D=22 S=35987     Ack=917551131 Seq=3100351282 Len=48 Win=8760
    desarrollo_IBM -> respaldo_IBM     RLOGIN R port=1023
    respaldo_IBM -> desarrollo_IBM     RLOGIN C port=1023 fabian\0fabian\0vt100/
    desarrollo_IBM -> respaldo_IBM     RLOGIN R port=1023
    desarrollo_IBM -> respaldo_IBM     RLOGIN R port=1023 Password:
    respaldo_IBM -> desarrollo_IBM     RLOGIN C port=1023
    respaldo_IBM -> desarrollo_IBM     RLOGIN C port=1023 d
    desarrollo_IBM -> respaldo_IBM     RLOGIN R port=1023 a
    respaldo_IBM -> desarrollo_IBM     RLOGIN C port=1023 d
    desarrollo_IBM -> respaldo_IBM     RLOGIN R port=1023 o
```

```
respaldo_IBM -> desarrollo_IBM    RLOGIN C port=1023
desarrollo_IBM -> respaldo_IBM    RLOGIN R port=1023 Login incorrect\r\n
respaldo_IBM -> desarrollo_IBM    RLOGIN C port=1023
desarrollo_IBM -> respaldo_IBM    RLOGIN R port=1023 login:
desarrollo_IBM -> respaldo_IBM    RLOGIN R port=1023
respaldo_IBM -> desarrollo_IBM    RLOGIN C port=1023 fabian\0fabian\0vt100/
desarrollo_IBM -> respaldo_IBM    RLOGIN R port=1023
desarrollo_IBM -> respaldo_IBM    RLOGIN R port=1023 Last login: Tue Aug
respaldo_IBM -> desarrollo_IBM    RLOGIN C port=1023
desarrollo_IBM -> respaldo_IBM    RLOGIN R port=1023 Sun Microsystems Inc
desarrollo_IBM -> respaldo_IBM    RLOGIN R port=1023 logout\r\n
desarrollo_IBM -> respaldo_IBM    RLOGIN R port=1023
respaldo_IBM -> desarrollo_IBM    RLOGIN C port=1023 fabian\0fabian\0vt100/
desarrollo_IBM -> respaldo_IBM    RLOGIN R port=1023 Last login: Tue Aug
respaldo_IBM -> desarrollo_IBM    RLOGIN C port=1023
respaldo_IBM -> desarrollo_IBM    RLOGIN C port=1023
desarrollo_IBM -> respaldo_IBM    RLOGIN R port=1023 logout\r\n
desarrollo_IBM -> respaldo_IBM    RSHELL R port=1023
respaldo_IBM -> desarrollo_IBM    RSHELL C port=1023 fabian\0fabian\0/usr/l
respaldo_IBM -> desarrollo_IBM    RSHELL C port=1023 C0644 0 hola.txt
respaldo_IBM -> desarrollo_IBM    RSHELL C port=1023 A4\n
```

En la salida anterior se puede identificar toda la información que se transmiten ambos servidores (respaldo_IBM y desarrollo_IBM). Aquí se va a almacenar absolutamente toda la información que fluya entre los dos, por lo que se recomienda filtrar los resultados que sean relevantes para revisar y poder detectar quien se está conectando, además de lo que está ejecutando. Esta información en teoría solo deberá arrojar conexiones del usuario backup y la ejecución de comandos para respaldar, por medio de servicios remotos (rlogin y rshell), aunque también podrá realizar conexiones con telnet o ftp. Por lo mencionado anteriormente se debe realizar un filtrado de los servicios de rlogin, rshell, ftp y telnet, mandarlo a un archivo y revisarlo semanalmente, como se muestra a continuación :

```
snoop -i salida.txt -t r > resulta.txt
egrep 'TELNET|FTP|RLOGIN|RSHELL' resulta.txt
```

Con lo anterior se obtiene una salida con las conexiones de telnet, ftp, rlogin y rshell, además de los comandos que se ejecutaron con estas.

Este monitoreo estará de manera permanente, ya que se supone que las conexiones entre ambos servidores son escasas, por eso la revisión del archivo de resultados debe ser sencilla, y en caso de que éste sea muy extenso, se debe atribuir a que algo raro se está suscitando.

Monitoreo remoto

Con el comando snoop sólo se va a monitorear a la plataforma IBM, sin embargo por tratarse éste de un servidor de monitoreo, faltaría agregar a las plataformas HP y Sun. Como un valor agregado al monitoreo de huecos de seguridad sobre las demás plataformas, se incluirá un monitoreo de funcionalidad de los servidores. Este será muy básico pero demasiado eficiente para detectar probables fallas que puedan afectar el ambiente de desarrollo y de producción. Consistirá de un constante envío de paquetes a todos los servidores (incluyendo al del Web) y a las bases de datos, sólo para detectar que estén respondiendo.

Para cada servidor habrá una revisión específica dependiendo de la funcionalidad de este. Este monitoreo arrojará información en línea, esto es, que se podrá detectar inmediatamente alguna falla o baja en un servidor.
ping *:* Este comando se encargará de verificar que los servidores respondan a la petición enviada, en caso de que lo haga, indica que por lo menos el servidor esta

encendido y con los servicios de red disponibles, en caso de que no responda a la petición del ping, indicará que hay graves problemas en el servidor, por que es indicio de que por lo menos los servicios de red no están disponibles o inclusive que el servidor esté dado de baja. También existe la posibilidad que la falla se deba a problemas de la red local (intranet).

El siguiente script va a verificar que todos los servidores respondan a la petición del ping y va a enviar un mensaje de alarma informando que servidor no respondió a dicha petición.

```
#!/usr/bin/ksh

cat /etc/hosts | cut -f1 > hosts.txt

for i in `cat hosts.txt`
do
     ping $i |cut -d" " -f3 >> result_ping_2.txt
done

paste hosts.txt result_ping_2.txt > result_3.txt
grep -v alive result_3.txt | cut -f1 > result_4.txt

for i in `cat result_4.txt`
do
write monitor <<EOT
El servidor $i no RESPONDE !!!!
EOT
done

rm result_ping_2.txt result_3.txt result_4.txt
```

Este script hace lo siguiente : primero saca la lista de los servidores que se encuentran en el /etc/hosts (aquí se encuentran los equipos de las plataformas HP, IBM y Sun), después se ejecuta un ping a cada uno y con estos resultados se determina si está arriba o abajo el servidor, si es el segundo caso, se envía un mensaje a la consola avisando inmediatamente al usuario que se usa para monitorear (monitor). Un ejemplo de la salida que enviaría en caso de que fallase algún servidor :

```
     Message from monitor on monitoreo_SUN (pts/2) [ Wed Jan 20 20:32:12 ] ...
El servidor desarrollo_IBM no RESPONDE !!!!
<EOT>

     Message from monitor on monitoreo_SUN (pts/2) [ Wed Jan 20 20:32:12 ] ...
El servidor produccion_HP no RESPONDE !!!!
<EOT>
```

Esto será suficiente para tomar inmediatamente las medidas necesarias.

tnsping : Este comando tiene la misma filosofía que el anterior, sólo que éste se usa a nivel de base de datos, esto significa que verifica que la base de datos responda, lo cual indicará que está activa.

El script verifica que las bases de datos de desarrollo_IBM y produccion_HP estén activas y respondan a esta petición, en caso contrario se enviará un mensaje de alarma notificando que alguna base de datos no responde. Este monitoreo se realiza por que con el comando ping solo podemos detectar si el servidor está funcionando, sin embargo, esto no garantiza que también la base de datos funcione correctamente.

```
#!/usr/bin/ksh

for i in `cat bdhosts.txt`
do
      tnsping $i |tail -1| cut -d" " -f1 >> res_ping_2.txt
done

paste bdhosts.txt res_ping_2.txt > res_3.txt
grep -v OK res_3.txt | cut -f1 > res_4.txt

for i in `cat res_4.txt`
do
write monitor <<EOT
La base de datos $i no RESPONDE !!!!
EOT
done

rm res_ping_2.txt res_3.txt res_4.txt
```

El script realiza lo siguiente : revisa una la lista que contiene las bases de datos que se desean revisar, en este caso existen 2 (la de desarrollo y la de producción), se ejecuta un tnsping a cada una y con estos resultados se determina si está arriba o abajo, si es el segundo caso, se envía un mensaje a la consola avisando inmediatamente al usuario que se usa para monitorear (monitor). Un ejemplo de la salida que enviaría en caso de que fallase algún servidor :

```
    Message from monitor on monitoreo_SUN (pts/2) [ Wed Jan 20 21:23:57 ] ...
La base de datos bd_desarrollo no RESPONDE !!!!
<EOT>
```

Esto será suficiente para tomar inmediatamente las medidas necesarias.

telnet al puerto 80 : El uso del comando telnet para monitorear puede parecer algo muy anticuado, sin embargo es una manera muy confiable de verificar que el puerto del servidor web esté respondiendo adecuadamente, y de esta manera detectar inmediatamente probables caídas de este y evitar que no se de servicio a nivel de internet.

```
#!/usr/bin/ksh

telnet web_HP 80 2> status.txt
VAR=`cat status.txt | cut -d" " -f2`

if [ $VAR = 'Unable' ]
then
write monitor <<EOT
El servidor WEB no RESPONDE !!!!
EOT
fi
rm status.txt
```

La función de este script es muy sencilla, por que solo realiza una verificación por medio de un telnet hacia el servidor web por el puerto 80, si éste no le responde a dicha petición, entonces se procede a enviar un mensaje, notificando esto :

```
Message from monitor on monitoreo_SUN (pts/2) [ Sat Jan 23 11:13:27 ] ...
El servidor WEB no RESPONDE !!!!
<EOT>
```

Con esto es necesario para hacer una verificación un poco sencilla, sin embargo de mucha utilidad para detectar y prevenir fallas en los servidores

5.4.3 Plataforma HP

En esta plataforma HP se encuentran los servidores de producción, esto significa, que debe ser la parte más segura de todo el entorno, por que aquí está la información importante. Existirá un servidor Web y otro servidor que tendrá la base de datos; en ambos servidores se instalarán las mismas herramientas de seguridad (figura 5.17), sólo que la configuración de cada una será diferente.

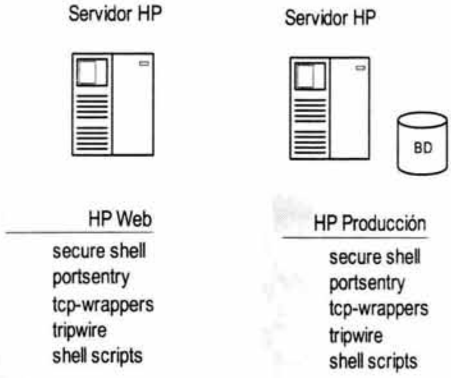


Figura 5.17 ESQUEMA DE SERVIDORES HP

5.4.3.1 Servidor Web

Secure shell

En el servidor web, secure shell se usará para que los usuarios se puedan conectar al servidor y elaboren las páginas web, además de la administración que se le debe de dar al web site. No sólo se podrán realizar documentos HTML directamente en el servidor, si no también se podrán ver con un browser abierto desde el secure shell, por medio del emulador gráfico. También se usará esta herramienta, para que los usuarios de este servidor puedan intercambiar información con la plataforma IBM, siendo el intermediario el servidor Sun de monitoreo (figura 5.18).

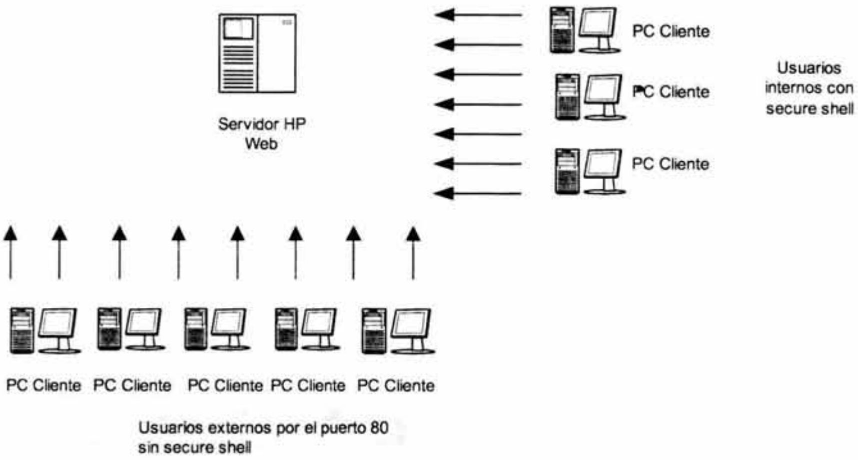


FIGURA 5.18 USO DE SECURE SHELL CON EL SERVIDOR WEB

Los parámetros que se van a modificar para este servidor son los siguientes :

VerboseMode	**yes**
Port	**60**
MaxConnections	**200**
Ciphers	**AnyCipher**
PermitEmptyPasswords	**No**
AllowUsers	**Lista de usuarios**
PermitRootLogin	**no**

Los valores que varían son el de número de conexiones (200 por tratarse de un servidor web) y el de permitrootlogin (para el usuario root sólo se pueda conectar desde una consola local o por medio del comando su).

Tcp-wrappers

Por medio de esta herramienta se hará un filtrado para que solamente se ocupen los servicios de red necesarios en este servidor. También se detectará de donde provienen los intentos de conexión fallidos hacia servicios de red que no estarán disponibles y con esto se obtendrá un mejor punto de referencia en caso de algún ataque. El servidor web que el administrador haya decidido instalar, debe tener sus propias bitácoras para verificar de donde se hacen peticiones hacia el servicio http, estas serán de mucha utilidad por que será un complemento a la bitácora de tcp-wrappers, la cual detecta los demás servicios de red. La configuración quedará de la siguiente manera, tomando en cuenta la figura 5.19 en donde se muestra cómo se conectarán a este servidor los usuarios internos y externos :

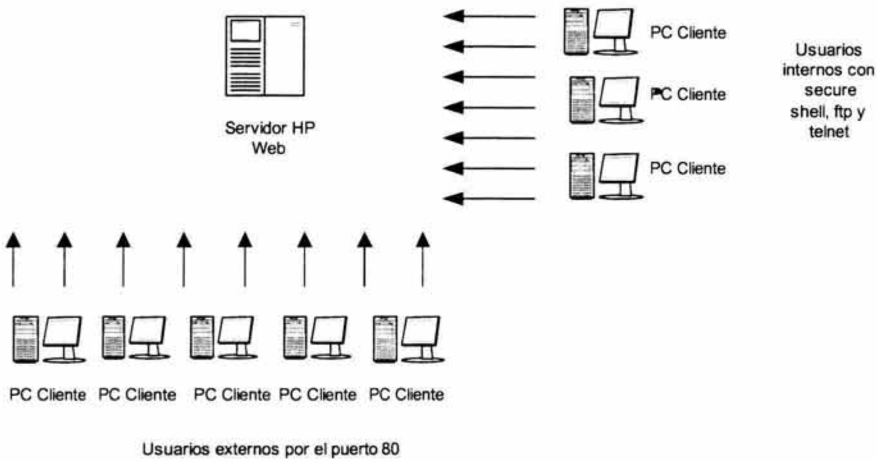


Figura 5.19 CONEXIÓN AL SERVIDOR WEB, LIMITADOS POR TCP-WRAPPERS

En el archivo /etc/inetd.conf estarán comentados todos los servicios de red, excepto los que se muestran en el cuadro, a los cuales se les asignará el demonio de tcp-wrappers para que capture en bitácora todos los intentos de conexión.

```
ftp     stream tcp nowait      root      /usr/lbin/tcpd             ftpd –l
telnet  stream tcp nowait root /usr/lbin/tcpd          telnetd
ssh     stream tcp nowait      root      /usr/lbin/tcpd sshd -i
finger  stream tcp nowait      bin       /usr/lbin/tcpd             fingerd
```

También se van a configurar los archivos /etc/hosts.allow y /etc/hosts.deny para determinar que equipos se podrán conectar a este servidor usando los servicios de red listados en el cuadro anterior.

El archivo /etc/hosts.allow, quedará de la siguiente manera :

```
ftpd, telnetd, sshd, fingerd : Lista de pc's que tendrán acceso
```

Lo anterior establece que le otorga el acceso a todos los equipos (direcciones IP separadas por coma) que aparezcan en esa lista. Para denegar los servicios restantes a todos los demás equipos, en el archivo /etc/hosts.deny poner lo siguiente :

```
ALL:ALL
```

Con lo anterior, en la bitácora de tcp-wrappers quedarán registrados todos los intentos de conexión hacia este servidor, para que se revise semanalmente. Se recomienda crear un directorio en donde se copien todas las bitácoras que se generen en este servidor, para que su revisión de estas sea más sencilla, utilizando scripts que muestren sólo lo que nos interesa encontrar.

Portsentry

El uso de esta herramienta estará enfocado primero en detectar que puertos se tienen habilitados, por que es muy común que un administrador no tenga esta información presente y ni siquiera sepa que huecos de seguridad puede tener en lo que a puertos se refiere. Para probar esta funcionalidad se recomienda instalar en cualquier servidor un software que haga un barrido de puertos a este servidor y obtener una lista inicial de los puertos que se están habilitando y descartar los que no se vayan a ocupar.

Otro uso que se le dará a esta herramienta, será la detección de barridos de puertos que se le hagan de manera intencional, por tratarse de un servidor web que dará servicio 24 x 7, entonces se deberá tener mucho cuidado con este intento de ataques, que podrían perjudicar la funcionalidad de éste.

Los puertos que se deben de monitorear son los que se usan menos y con poca frecuencia, en este caso no se debe monitorear el puerto 80 debido a que es el que regularmente se utiliza para los servidores Web, aunque se puede utilizar cualquier otro.

En el archivo de configuración portsentry.conf, se deben quitar los símbolos de comentario a las especificaciones de los puertos (TCP y UDP), los cuales se encuentran en las opciones UDP_PORTS y TCP_ PORTS :

```
TCP_PORTS=" 1,7,9,11,13,17,19,20,21,22,23,25,37,43,53,77,79,87,95,101,102,109, 110,
111,113,115,117,119, 137,138,139,152,179,382,383,512,513,514,515,520,526,530, 531,
532,540,543,544,545,556,749,750,751,754,760,761,987,1109,1110,1111,1260,1476, 1508,
1524,1536,1553,1712,1788,1889,2049,2105,2106,3275,4444,4672,4789,5300,5301,5302,
5303,5304,5305,5403,5404,5408,5555,5696,5707,5709,5710,5999,6111,6868,6874,6875,
7489,7815,8868,9999,10000,31766"

UDP_PORTS="7,9,13,19,22,37,39,53,67,68,69,88,111,123,137,138,139,161,162,512,513,
514,517,518,520,525,533,570,749,750,1067,1068,1110,1553,1591,1683,1686,1712,1744,
1788,1889,2049,5302,5303,5708"
```

Como se instaló TCP-Wrappers en este servidor se deberá quitar el símbolo de comentario a la línea de KILL_HOSTS_DENY :

```
KILL_HOSTS_DENY="ALL: $TARGET$"
```

Cuando suceda un barrido de puertos se ejecutará un shell script que mande un correo al administrador del servidor web, notificándole que están ejecutando un barrido de puertos en ese servidor.

```
KILL_RUN_CMD=" mailweb.sh "
```

El contenido del shell script es el siguiente :

```
#!/usr/bin/ksh

mailx -s barrido_puertos webadmin <<EOT
Se esta ejecutando un barrido de puertos en este servidor
EOT
```

El administrador del servidor web, recibirá un mensaje como éste :

```
Message 93:
From root Sat Jan 6 14:28:34 CDT 2003
Received: (from root@localhost)
    by web_HP(8.9.3 (PHNE_24419)/8.8.6) id OAA12611
    for webadmin; Sat, 6 Jan 2003 14:28:34 -0500 (CDT)
Date: Sat, 6 Jan 2003 14:28:34 -0500 (CDT)
From: root
Message-Id: <200309061928.OAA12611@prod_sis>
To: webadmin
Subject: barrido_puertos
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Status: R

Se esta ejecutando un barrido de puertos en este servidor

?
```

Con esta configuración se puede controlar el barrido de puertos a través de un monitoreo en línea, sin embargo, también se recomienda revisar las bitácoras semanalmente.

Tripwire

Para este servidor en particular, se usará esta herramienta de seguridad para que revise constantemente los archivos que se utilizan para el servidor web, tanto los archivos de configuración como los archivos que contienen las páginas de este web site. Se deberá ejecutar semanalmente por las constantes consultas y cambios que se le puedan hacer a estos archivos. Se recomienda también revisar los archivos del sistema operativo.

El archivo tw.config tendrá el contenido que a continuación se lista :

```
## Archivos del Sistema operativo

/.rhosts            R
/.profile           R
/.cshrc             R
/.login             R
/.forward           R
/etc/inetd.conf     R
/etc/exports        R
/etc/utmp           L
/etc/group          R
/etc/passwd         L
/etc/diskinfo       R
/etc/dmesg          R
/etc/dump           R
/etc/ping           R
/etc/wall           R
/etc/arp            R
/etc/hosts          R
/etc/inittab        R
/etc/fstab          R
/etc/timezone       R
/etc/profile        R
/etc/rc.log         R
/etc/services       R
/etc/resolv.conf    R
/etc/netgroup       R
/etc/networks       R
/etc/nsswitch.conf  R
/etc/hosts.equiv    R
```

```
/etc/exportfs          R
/etc/securetty         R
/etc/utmp              R
/etc/syslog.conf       R
/etc/mnttab            R
/etc/ftpd/ftpusers     R
/etc/lvmtab            R
/etc/shutdown.allow    R
/etc/pfs_fstab         R
/stand/vmunix          R
/bin/login             R
/bin/ipcs              R
/bin/passwd            R
/bin/su                R
/bin/write             R
/usr/bin/at            R
/usr/bin/bdf           R
/usr/bin/crontab       R
/usr/bin/iostat        R
/usr/bin/mailx         R
/usr/bin/netstat       R
/usr/bin/rcp           R
/usr/bin/rlogin        R
/usr/bin/remsh         R
/usr/bin/vmstat        R
/var/adm/syslog/syslog.log    R
/var/adm/inetd.sec            R
/var/adm/sulog                R
/var/adm/wtmp                 R
/var/adm/btmp                 R

## Paginas y graficas del Webserver que no cambian constantemente

$WEB_SITE/gif/*
$WEB_SITE/*html
$WEB_SITE/*htm

## Binarios del Webserver

$WEB_SERVER/bin/*

## Archivos de configuración del Webserver

$WEB_SERVER/conf/*
```

Con el archivo de configuración anterior se debe compilar la herramienta de tripwire, para que verifique los archivos más importantes de este servidor, sobretodo los que corresponden al web Server. Esta bitácora también deberá moverse al directorio mencionado en párrafos anteriores.

Monitoreo local

Como este servidor es muy crítico, es necesario monitorear los elementos que puedan afectar su buen funcionamiento como : el procesador, la memoria y el disco.

Rendimiento de CPU

El monitoreo hacia el procesador es indispensable para detectar como se está comportando, sobretodo por los intentos de conexión hacia el web site. Se utilizará el comando sar para ver el uso del cpu.

```
server1:>sar 1 5

20:18:31  %usr  %sys  %wio  %idle
20:18:32   28    29    0     44
20:18:33   17    41    1     42
20:18:34   21    45    0     34
20:18:35   34    55    0     11
```

```
20:18:36   19   38   0   43

Average    24   41   0   35
```

El monitoreo se realizará a través del siguiente shell script :

```
#!/usr/bin/ksh

sar 300 3 > res_sar.txt

VAR=`grep Average res_sar.txt | tr -s " " " "| cut -f5 -d" "`

if [ VAR -le 10 ]
then
write webadmin <<EOT
El CPU del servidor Web esta a mas del 90% de su uso
EOT
fi
```

El comando sar toma tres muestras en un lapso de 15 minutos (sar 300 3), el resultado obtenido lo manda a un archivo, el cual será filtrado para sacar el porcentaje total del CPU que esta libre, si este umbral es menor a 10, mandará un mensaje en la consola para que el administrador del web server o el administrador del servidor verifiquen que está afectando el performance del servidor.

```
Message from root (pts/2) [ Tue Jan 09 21:27:56 ] ...
El CPU del servidor Web esta a mas del 90% de su uso
<EOT>
```

Este script deberá ejecutarse por medio de un crontab cada 15 minutos.

Memoria

La memoria también juega un papel muy importante en el desempeño del servidor web, por eso se debe de cuidar su porcentaje de uso, para que en caso de que exceda umbrales críticos, se tomen medidas correctivas antes de que afecte el performance del servidor. Con el comando vmstat se obtendrá esta información para prevenir la saturación de la memoria o el incremento de memoria swap.

```
server1:> vmstat
   procs      memory            page                  faults     cpu
 r  b  w    avm   free  re  at  pi  po  fr  de  sr  in   sy   cs us sy id
 1  0  0  21282 217571 119  26   0   0  34   0   0 419 14645 657 21 41 38
```

La revisión de memoria se podrá realizar por medio de un cron que ejecute el siguiente shell script, cada 15 minutos :

```
vmstat | egrep -v 'memory|avm' | tr -s "    " " " | cut -d" " -f6 > /tmp/vm2

 for x in `cat /tmp/vm2`
  do
   let dispo=x*4
   let dispo1=dispo/1024
  done
      echo La memoria RAM disponible en este servidor es de $dispo1 MB
  total=2048
  let promedio=dispo1*100
  let promedio1=promedio/total
  let prom=100-promedio1
       echo Se esta utilizando el $prom % de la memoria RAM
 if [ prom -ge 90 ]
 then
```

```
    write webadmin <<EOT
    El CPU del servidor Web está a mas del 90% de su uso
    EOT
    fi
```

Con este script se obtiene la memoria RAM que tiene disponible el servidor y el porcentaje que está utilizando. En caso de que este porcentaje rebase el 90% de utilización, enviará mensaje a la consola, para advertir de este suceso.

```
        Message from root (pts/2) [ Wed Jan 10 14:11:10 ] ...
    El CPU del servidor Web esta a mas del 90% de su uso
    <EOT>
```

Memoria swap

Como auxiliar a la memoria física, esta la memoria swap, la cual también puede agotarse, por eso es conveniente monitorearla al igual que la física. Con este tipo de memoria no hay tanto problema por que se puede ir agregando conforme se vaya requiriendo, siempre y cuando haya espacio libre en el disco. En este servidor la memoria swap se va a monitorear con el comando swapinfo (comando propio de HP-UX), para detectar como va incrementándose y si es necesario, aumentarle su capacidad.

También la memoria swap será revisada por medio de un cron que ejecute el siguiente shell script, cada 30 minutos :

```
swapinfo | egrep -v 'Kb|TYPE|reserve|memory' | tr -s "       " " "| cut -d" " -f5 | cut -d"%" -f1 > info.txt

    for x in `cat info.txt`
    do
        if [ $x -ge 90 ]
        then
        echo 1 > status.txt
        else
        echo 0 > status.txt
        fi
    done

    VAR=`cat status.txt`

    if [ $VAR = 0 ]
    then
    exit
    else
    write webadmin <<EOT
    El swap esta a mas del 90%
    EOT
fi
```

Con este script se obtiene el porcentaje de memoria swap que se está utilizando, si este sobrepasa el 90% entonces enviará un mensaje a la consola, advirtiendo de este problema, para que sea corregido inmediatamente agregando más swap en el disco.

El mensaje que envía, es el siguiente :

```
    Message from root (pts/2) [ Wed Jan 10 20:13:30 ] ...
    El swap esta a mas del 90%
    <EOT>
```

Cabe mencionar que los scripts anteriores deberán ser ejecutados por el usuario root, debido a los comandos que se ocupan para obtener la información.

Espacio en disco :

Por tratarse de un servidor web, se debe tener especial cuidado en los filesystems donde se alojen las páginas del website. Estos directorios se componen de páginas html, gráficos y en ocasiones videos, lo que ocasiona que exista el riesgo de la saturación de espacio, por eso se recomienda vigilar constantemente este servidor.

bdf : Por tratarse de un servidor HP se usará el comando bdf, que arroja la misma salida que un df –k en cualquiera de los otros servidores, esto quiere decir que se obtendrá la misma salida. En este script se hace un filtrado obteniendo los valores que sobrepasan el 85%, entonces se manda un mensaje a consola avisando de este suceso (figura 5.20), para determinar que tan crítico es el Filesystem que se está llenando.

```
#!/usr/bin/ksh

bdf | grep -v used | tr -s "      " " " | cut -f5,6  -d" " | tr -s "%" " " > cuatro

awk '$1 > 85 {print $1 $2}' cuatro > resulta.txt

VAR=`ll resulta.txt | tr -s "   " " " | cut -f5 -d" "`

if [ VAR -eq 0 ]
then
exit
else
write webadmin <<EOT
Algunos filesystems sobrepasan el 85% de su uso, verificarlo en el archivo : resulta.txt
EOT
fi
```

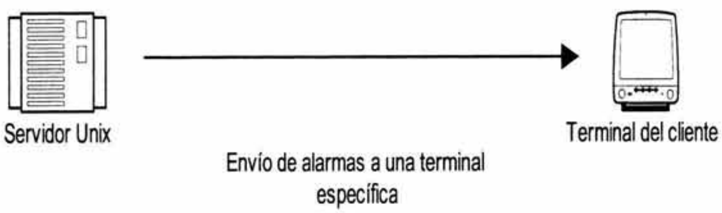


Figura 5.20 NOTIFICACIÓN DE ALARMAS

5.4.3.2 Servidor de base de datos

Secure shell

Para este servidor se requiere de secure shell para que los usuarios de la base de datos se puedan conectar por medio de esta herramienta de seguridad. Es muy común que cualquier base de datos no maneje elementos de seguridad, por lo que se requiere primero que se realice una conexión al servidor con un usuario de Unix y después la conexión a la base de datos con el usuario de este manejador. Con lo anterior se tiene la certeza de que la contraseña no viajará en claro a través de la red.

Con esta herramienta también se podrán ejecutar aplicaciones gráficas que accedan a la base de datos. La mayoría de los parámetros quedarán con su valor de default excepto los siguientes:

VerboseMode	yes
Port	70
MaxConnections	100
Ciphers	AnyCipher
PermitEmptyPasswords	No
AllowHosts	Lista de pc's
AllowUsers	Lista de usuarios
PermitRootLogin	No

En esta lista los parámetros que cambiaron son : el puerto de conexión, ya que para tener un mejor control se ocupará uno que no esté reservado por otro servidor, el de número de conexiones, 100 por tratarse del servidor de producción sólo se deberán conectar los usuarios indispensables y el de permitrootlogin (para el usuario root sólo se pueda conectar desde una consola local o por medio del comando su).

Tcp-wrappers

El uso de esta herramienta en el servidor de base de datos, permitirá detectar cualquier intento de intrusión al servidor, tomando en cuenta que este servidor es el más delicado, por la información tan importante que va a manipular. Se hará un filtrado de los servicios de red que se vayan a ocupar para listarlos en el archivo de configuración y así poder denegar los accesos a los demás servicios de red innecesarios.

En el archivo /etc/inetd.conf no estarán comentados los siguientes servicios de red a los que se les asignará el demonio de tcp-wrappers para que capture en bitácora todos los intentos de conexión. Se habilitarán los servicios de red remotos sólo a los servidores que los necesiten, ya que con tcp-wrappers se puede restringir el uso de estos.

ftp	stream	tcp	nowait	root	/usr/lbin/tcpd	ftpd -l
telnet	stream	tcp	nowait	root	/usr/lbin/tcpd	telnetd
ssh	stream	tcp	nowait	root	/usr/lbin/tcpd	sshd -i
finger	stream	tcp	nowait	bin	/usr/lbin/tcpd	fingerd
login	stream	tcp	nowait	root	/usr/lbin/tcpd	rlogind
shell	stream	tcp	nowait	root	/usr/lbin/tcpd	remshd
exec	stream	tcp	nowait	root	/usr/lbin/tcpd	rexecd

En este servidor se van a configurar los archivos /etc/hosts.allow y /etc/hosts.deny para permitir o denegar conexiones al servidor. El archivo /etc/hosts.allow, quedará de la siguiente manera:

```
ftpd, telnetd, sshd, fingerd : Lista de pc's que tendrán acceso
rlogind, remshd, rexecd : Lista de servidores que tendrán acceso
```

Para denegar los servicios restantes a todos los demás equipos, el archivo /etc/hosts.deny quedará de así:

```
ALL:ALL
```

La bitácora de tcp-wrappers deberá revisarse cada semana, también se recomienda tener un directorio exclusivo de bitácoras para su revisión.

Portsentry

En un servidor donde se aloje una base de datos, es muy importante detectar si se ejecutan barridos de puertos, sobretodo por que cuando un servidor cuenta con una base de datos, los clientes hacen peticiones a más de un puerto. Otro punto que no debe pasarse por alto, es que cualquier base de datos difícilmente tiene herramientas de seguridad que la protejan, por esto se deben buscar herramientas auxiliares que cubran estas carencias.

A la configuración de la herramienta portsentry se le hará lo siguiente, se le quitarán los símbolos de comentario a las especificaciones de los puertos TCP y UDP.

```
TCP_PORTS=" 1,7,9,11,13,17,19,20,21,22,23,25,37,43,53,77,79,87,95,101,102,109, 110,
111,113,115,117,119, 137,138,139,152,179,382,383,512,513,514,515,520,526,530, 531,
532,540,543,544,545,556,749,750,751,754,760,761,987,1109,1110,1111,1260,1476, 1508,
1524,1536,1553,1712,1788,1889,2049,2105,2106,3275,4444,4672,4789,5300,5301,5302,
5303,5304,5305,5403,5404,5408,5555,5696,5707,5709,5710,5999,6111,6868,6874,6875,
7489,7815,8868,9999,10000,31766"

UDP_PORTS="7,9,13,19,22,37,39,53,67,68,69,88,111,123,137,138,139,161,162,512,513,
514,517,518,520,525,533,570,749,750,1067,1068,1110,1553,1591,1683,1686,1712,1744,
1788,1889,2049,5302,5303,5708"
```

En este servidor también se tiene instalada la herramienta TCP-Wrappers, por lo tanto, se deberá quitar el símbolo de comentario a la línea de KILL_HOSTS_DENY :

```
KILL_HOSTS_DENY="ALL: $TARGET$"
```

Cuando suceda un barrido de puertos se ejecutará un shell script que mande un correo al usuario root, notificándole de este suceso.

```
KILL_RUN_CMD=" mailbd.sh "
```

El contenido del shell script es el siguiente :

```
#!/usr/bin/ksh

mailx -s barrido_puertos bdaminOT
Se esta ejecutando un barrido de puertos en este servidor
EOT
```

Con esta configuración se puede controlar el barrido de puertos a través de un monitoreo en línea, sin embargo, también se recomienda revisar las bitácoras semanalmente.

Tripwire

En este servidor la herramienta de tripwire revisará los archivos de configuración de la base de datos y descartara los archivos donde se almacena la información de ésta, ya que va cambiando constantemente, por lo tanto, no tiene caso que se revise. Por tratarse de un servidor crítico, también se revisarán los archivos de configuración del sistema operativo. La ejecución de la herramienta se llevará a cabo cada tercer día, reportando solamente los cambios realizados.

El archivo de configuración tw.config quedará así:

```
## Archivos del Sistema operativo

/.rhosts                R
/.profile               R
/.cshrc                 R
/.login                 R
/.forward               R
/etc/inetd.conf         R
/etc/exports            R
/etc/utmp               L
/etc/group              R
/etc/passwd             L
/etc/diskinfo           R
/etc/dmesg              R
/etc/dump               R
/etc/ping               R
/etc/wall               R
/etc/arp                R
/etc/hosts              R
/etc/inittab            R
/etc/fstab              R
/etc/timezone           R
/etc/profile            R
/etc/rc.log             R
/etc/services           R
/etc/resolv.conf        R
/etc/netgroup           R
/etc/networks           R
/etc/nsswitch.confR
/etc/hosts.equiv        R
/etc/exportfs           R
/etc/securetty          R
/etc/utmp               R
/etc/syslog.conf        R
/etc/mnttab             R
/etc/ftpd/ftpusers R
/etc/lvmtab             R
/etc/shutdown.allow     R
/etc/pfs_fstab          R
/stand/vmunix           R
/bin/login              R
/bin/ipcs               R
/bin/passwd             R
/bin/su                 R
/bin/write              R
/usr/bin/at             R
/usr/bin/bdf            R
/usr/bin/crontab        R
/usr/bin/iostat         R
/usr/bin/mailx          R
/usr/bin/netstat        R
/usr/bin/rcp            R
```

```
/usr/bin/rlogin            R
/usr/bin/remsh             R
/usr/bin/vmstat            R
/var/adm/syslog/syslog.log R
/var/adm/inetd.sec         R
/var/adm/sulog             R
/var/adm/wtmp              R
/var/adm/btmp              R

## Archivos de configuración de la base de datos

$BD_HOME/admin/*

## Binarios de la base de datos

$BD_HOME/bin/*
```

Con este archivo se va a compilar la herramienta de tripwire para que cada se revisen los archivos del sistema operativo y los de base de datos los cuales son muy trascendentales para la funcionalidad de esta entidad gubernamental.

Monitoreo local

La función de este servidor es la más importante en este esquema, por eso se debe realizar un monitoreo constante al procesador, a la memoria y al disco, a este ultimo se le debe poner más atención por la constante escritura que la base de datos realiza.

Rendimiento de CPU

En un servidor de base de datos el acceso al procesador es muy relevante, por que a menudo se tiene que bajar la información alojada en la memoria hacia el disco. El comando sar se encargará de monitorear el CPU, por medio de este script:

```
#!/usr/bin/ksh

sar 300 3 > res_sar.txt

VAR=`grep Average res_sar.txt | tr -s " " " "| cut -f5 -d" "`

if [ VAR -le 10 ]
then
write bdadmin <<EOT
El CPU del servidor Web esta a mas del 90% de su uso
EOT
fi
```

La funcionalidad de este script es igual que la del servidor anterior, esto es, se tomarán 3 muestras en un lapso de 15 minutos (sar 300 3), el resultado se va a un archivo temporal, el cual es filtrado para obtener el porcentaje total del CPU que está libre, si este umbral es menor a 10, se enviará un mensaje a la consola dirigido al administrador de la base de datos.

Memoria

Como se mencionó en el punto anterior, la base de datos ocupa constantemente la memoria, aparte de la que reserva al momento de levantar la base de datos, por eso es indispensable revisar su comportamiento. Este monitoreo se hará con el comando vmstat, para detectar el espacio libre.

La revisión de memoria se hará igual que en el servidor webHP por medio de un cron que ejecute el siguiente shell script (cada 15 minutos):

```
vmstat | egrep -v 'memory|avm' | tr -s "    " " " | cut -d" " -f6 > /tmp/vm2

 for x in `cat /tmp/vm2`
  do
   let dispo=x*4
   let dispo1=dispo/1024
  done

  echo La memoria RAM disponible en este servidor es de $dispo1 MB

  total=2048
  let promedio=dispo1*100
  let promedio1=promedio/total
  let prom=100-promedio1

  echo Se está utilizando el $prom % de la memoria RAM

if [ prom -ge 90 ]
then
write bdadmin <<EOT
El CPU del servidor de Base de datos está a más del 90% de su uso
EOT
fi
```

Si el porcentaje rebase el 90% de utilización, enviará mensaje a la consola, para advertir de este suceso.

Memoria swap

Como se trata de un servidor de base de datos, la memoria swap es trascendental para su óptimo funcionamiento. El monitoreo de la memoria swap se hará a través del comando swapinfo, este deberá ser muy constante, por que la falta de esta memoria le afecta notablemente a la base de datos y, por consiguiente, a la respuesta que tenga hacia con los usuarios.

La memoria swap será revisada por medio de un cron cada 30 minutos, el cual ejecutará el siguiente script:

```
swapinfo | egrep -v 'Kb|TYPE|reserve|memory' | tr -s "     " " "| cut -d" " -f5 | cut -d"%" -f1 >
info.txt

for x in `cat info.txt`
do
     if [ $x -ge 90 ]
     then
     echo 1 > status.txt
     else
     echo 0 > status.txt
     fi
done

VAR=`cat status.txt`

if [ $VAR = 0 ]
then
exit
else
write bdadmin <<EOT
El swap esta a mas del 90%
EOT
fi
```

Si el porcentaje de uso del swap sobrepasa el 90% entonces enviará un mensaje a la consola, para que se agregue más swap al servidor y con esto prevenir fallas.

Espacio en disco

En esta parte se revisarán los filesystems que contengan los archivos de la base de datos y obviamente del sistema operativo.

bdf : El espacio que se le asigna a la base de datos es muy relevante, ya que la carencia de este, puede ser motivo de que la base de datos quede fuera de línea o que se corrompa la información que almacena. Por lo tanto, es importante monitorear los filesystems que contienen los archivos de la base datos para detectar como van incrementándose, así como el filesystem del sistema operativo, definiendo un umbral crítico a partir del 85% y enviando un mensaje a consola cuando esto suceda.

```
#!/usr/bin/ksh

bdf | grep -v used | tr -s "      " " " | cut -f5,6  -d" " | tr -s "%" " " > cuatro

awk '$1 > 85 {print $1 $2}' cuatro > resulta.txt

VAR=`ll resulta.txt | tr -s "   " " " | cut -f5 -d" "`

if [ VAR -eq 0 ]
then
exit
else
write bdadmin <<EOT
Algunos filesystems sobrepasan el 85% de su uso, verificarlo en el archivo : resulta.txt
EOT
fi
```

5.4.3.3 Servidor de pruebas

Es difícil que en un área de trabajo exista un servidor que solo se ocupe para realizar pruebas y posteriormente aplicar estos cambios en un ambiente de producción, sin embargo, es muy recomendable ya que los problemas que se vayan suscitando, no le afectan a un servidor crítico, ni se pone en riesgo la seguridad que es principalmente lo que se esta buscando con esta tesis. Cuando no se pueda contar con un servidor de pruebas, entonces se pueden aplicar cambios a un servidor que no sea muy critico, tal vez a uno de desarrollo. El servidor que en esta tesis se ocupará como de pruebas será el servidor de desarrollo

Revisión de bitácoras

Es conveniente realizar una verificación constante a las bitácoras, que son las principales fuentes de información de los problemas que presente un servidor (los shell scripts mandan todas sus salidas a bitácoras para su revisión semanal o mensual). En las bitácoras está de manera detallada todo lo que ha sucedido con el servidor, como intentos de conexión al servidor, cuando se da de baja y cuando se inicia el servidor, fallas en dispositivos (hardware), problemas con la memoria, problemas con parámetros del kernel, problemas con el procesador, espacio en disco etc., Solo basta emplear algunas utilerías de unix para facilitar la tarea de revisión, ya que son archivos demasiado grandes y no es factible verlos con un editor de texto, entonces bastará con un filtrado de palabras claves sobre la bitácora y mandar la información a un archivo en caso de encontrar algo anormal.

Las bitácoras se enviarán hacia el servidor donde estará el repositorio de bitácoras (servidor Sun de monitoreo), por medio de un ftp seguro o también se pueden utilizar los servicios de red remotos, para la transferencia de estos archivos (figura 5.21). La ventaja de la mayoría de los shell scripts es que mandan a consola un mensaje de notificación, por lo tanto estas no es necesario revisarlas.

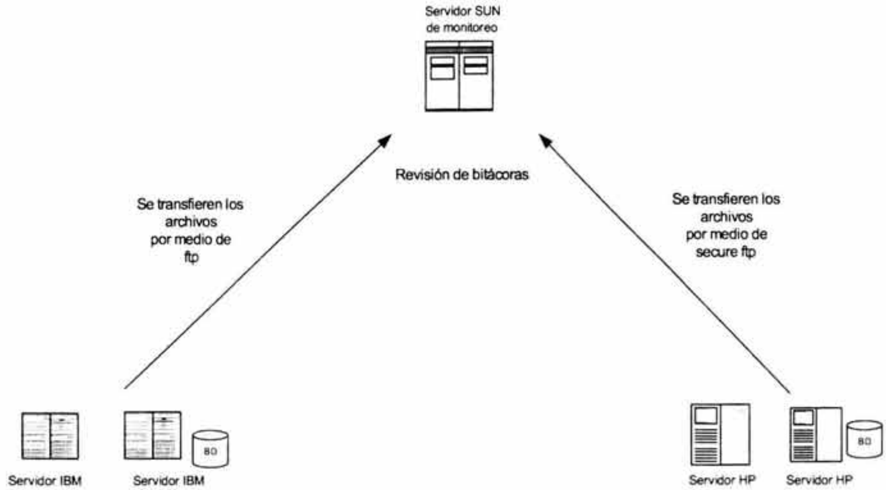


Figura 5.21 ENVIÓ DE BITÁCORAS

El esquema completo incluyendo el entorno protegido se muestra en la figura 5.22 :

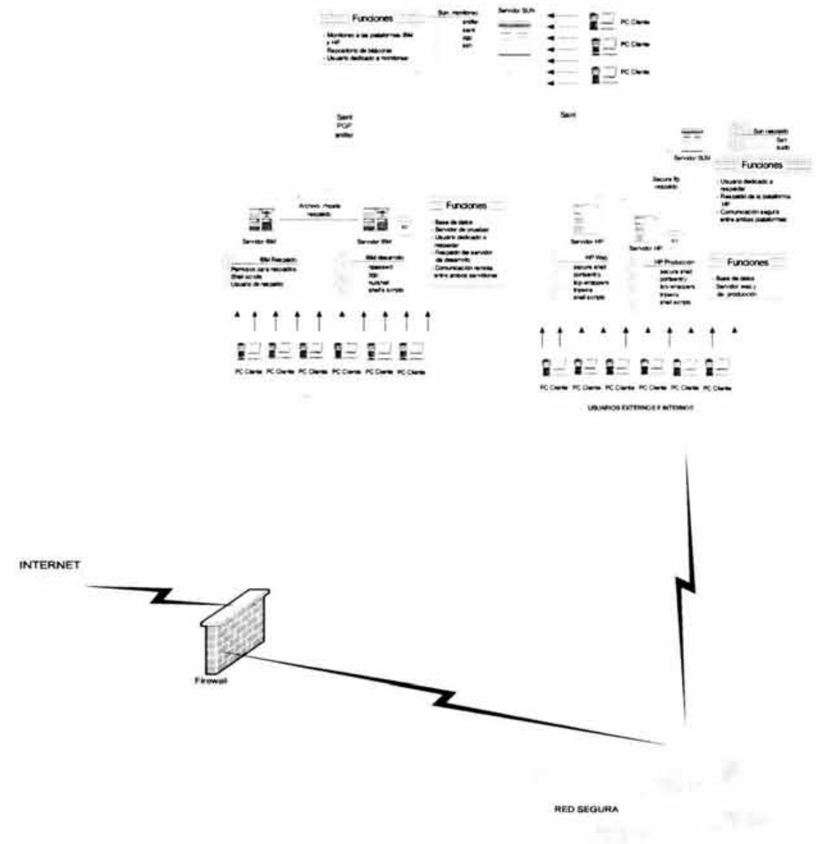


Figura 5.22 ESQUEMA COMPLETO INCLUYENDO FIREWALL Y RED SEGURA

6. CONCLUSIONES

Terminada la elaboración de la tesis "Propuesta de integración de un esquema de seguridad en el Centro de Cómputo de una Entidad Gubernamental, utilizando Herramientas de Libre Distribución" se puede concluir que las herramientas de libre distribución son una magnifica opción para toda entidad o empresa que desee tener seguridad en sus servidores y en su centro de cómputo, ya que brinda grandes ventajas sobre las herramientas de seguridad comerciales, aun cuando se requiera un poco más de tiempo en el diseño del esquema a implantar (en esta tesis se propusieron tres), así como un poco de complejidad al instalarlas y configurarlas.

El propósito de esta tesis se cumple desde el inicio al dar una introducción de un tema muy difícil de manejar como es la seguridad a nivel de sistema operativo Unix, principalmente para saber de principio que es lo que el administrador del sistema debe de arreglar en éste, antes de proseguir con los demás temas. En el segundo tema ya se toca la base de esta tesis, que son las herramientas de seguridad de libre distribución, de las que se explica su funcionalidad y las características que tienen, a continuación se detalla su instalación y configuración, lo que vendría ser la columna vertebral de la tesis, por que de esto dependerá el tipo de esquema que se desee implantar en la entidad o empresa.

Posteriormente se hace un análisis de la problemática actual de la entidad, la cual esta basada en la desventaja que tiene el entorno desprotegido contra el entorno seguro, ya que para éste se adquirió toda una suite de software de seguridad, la cual la implantó personal especializado en éste software, además de otorgar capacitación para su uso y su administración. Cabe mencionar que cada producto instalado requería grandes espacios de almacenamiento, en algunos casos era necesario un servidor exclusivo para que no afectará el "performance" de algún servidor de producción.

A lo anterior, cabe mencionar que la suite de seguridad instalada en la red segura no es flexible ya que sólo se puede instalar en ciertas plataformas y en otros casos no había compatibilidad entre el software y hardware, por lo tanto, se considera una restricción muy importante, además que, para monitorear servidores o equipos dentro del entorno, es necesario la instalación de agentes en cada uno de estos. Estos agentes son procesos corriendo en cada uno de los servidores, por lo tanto, cuando hay alguna actualización del software se requiere distribuir los agentes en cada uno.

Cada uno de estos productos pueden ser usados casi de la misma manera que los de libre distribución, ya que el objetivo de ambos es mantener seguros y vigilados los servidores que están a su alrededor, sin embargo existen grandes desventajas entre los productos comerciales y los productos de libre distribución.

Ventajas de las herramientas de libre distribución sobre las comerciales

- Una de las más importantes es el costo, mientras éste es gratuito el otro tiene un costo muy elevado.
- La facilidad de su instalación, uso y administración sin la necesidad de cursos largos y de costos elevados.
- La experiencia de varios administradores de servidores Unix que los usan actualmente y que se han enfrentado a problemas o detalles con estas

herramientas mientras que los comerciales requieren de soporte que en algunas ocasiones no viene incluido al comprar el producto.

- La documentación que existe en internet acerca de las herramientas de libre distribución es mucho mayor que la que existe para las herramientas comerciales.
- Estas pueden ejecutarse en un mismo servidor sin pegarle al performance.
- Se pueden instalar casi en todas las plataformas Unix.
- No se requiere de contratar personal especializado para su uso.
- Puede convivir con las herramientas comerciales.

El último punto es el más importante para el caso de esta tesis, por que en la entidad ya existen herramientas comerciales en algunos servidores, sin embargo, este no es un inconveniente para poder implantar esquemas con las herramientas de libre distribución y que puedan convivir en un mismo entorno, si es necesario. Entonces la solución más viable fue instalar herramientas competitivas o de mayor nivel que las comerciales que satisfagan a los usuarios finales, sin darle complicaciones y adaptándolas a sus necesidades.

Es muy importante el impulso que se le da en esta tesis a los administradores, para poner mayor énfasis en la seguridad de los servidores que administran, esto debido a la gran demanda que hay actualmente con el uso de estos, por tal motivo se elaboraron capítulos de instalación y configuración de las herramientas de seguridad para otorgarles facilidad en algo que podría convertirse tan complejo y a la postre origine que no se lleve a cabo la implantación de uno o más esquemas.

Esta tesis juega un papel muy importante en la decisión de implantar un esquema de seguridad en una entidad o empresa, ya que tiene las herramientas necesarias para que se lleve a cabo esta implantación en poco tiempo y sin la necesidad de tener una vasta experiencia en la administración del sistema operativo Unix.

Se recomienda distribuir esta tesis a los administradores de Unix de cualquier plataforma, para que fortalezcan las bases de seguridad con las que cuentan, si fuera el caso, si no para que comiencen a adentrarse mas con éste tema tan importante, el cual, se puede aplicar no solo en las grandes empresas o entidades, si no en cualquier lugar donde se cuente con un servidor Unix.

7. Bibliografía

- Tsutomu Shimomura y John Markoff. *Takedown,* Editorial El Pais-Aguilar.
- David A. Curry. *Unix Systems Programming,* O'Reilly.
- Martin Otero, Martín Morales y Pablo Romo. *Administración Unix,* Fideicomiso SEP-UNAM.
- Evi Nemeth, Gart Snyder, Scott Seebas y Trent Hein. *Unix System Administration,* Prentice-Hall.
- Kirk Waingrow. *Unix, Tips Y Trucos De Administracion,* Pearson
- AEleen Frisch. *Essential System Administration*
- Daniel J. Barrett y Richard Silverman. *SSH, The Secure Shell.*
- Jerry D. Peek, Tim O'Reilly y Mike Loukides. *UNIX Power Tools.*
- W. Richard Stevens. *Advanced Programming in the Unix Environment.*
- Página que distribuye el software de openssh
  http://www.openssh.com/
- Página de secure shell (comercial)
  http://www.ssh.com/
- Seguridad en cómputo de la UNAM.
  http://www.asc.unam.mx/
- Centro de Seguridad en Internet.
  http://www.cert.org/
- Interconexión de los Recursos Informáticos de las universidades
  http://www.rediris.es/
- Equipo de Respuesta a Incidentes de Seguridad en Cómputo
  http://www.unam-cert.unam.mx/

A. Glosario

"Bug" : traducido literalmente como "bicho", adquiere otro significado cuando hablamos de informática, se refiere a elementos y circunstancias en el software o hardware, involuntarios e indeseados, que provocan un malfuncionamiento.

Shutdown : Se utiliza este termino cuando se va a apagar o a dar de baja un servidor.

Reboot : Este termino es utilizado comúnmente al reiniciar un servidor unix

Shell : Programa en unix que permite comprender los comandos, tiene 3 usos principales : interactuar con el usuario, personalizar sesiones de unix y la programación.

shell-scripts : Son programas realizados con una serie de comandos especiales del sistema operativo unix (programación shell).

Timeout : Este termino se ocupa cuando se termina el tiempo de espera para poder establecer una conexión.

IP spoofing : Técnica usada para poder obtener acceso (sin autorización) a servidores, enviando paquetes con direcciones IP aparentemente válidas.

ICMP (Internet Control Message Protocol) : Este protocolo soporta paquetes que contienen errores, campos de control y mensajes con información. El comando ping usa ICMP para verificar una conexión.

Demonio : Es un proceso que corre en background y ejecuta una operación específica. Regularmente este proceso espera ser llamado por algún programa.

Background : Son procesos que se pueden ejecutar en segundo plano, esto es, no interactúan con el usuario, pero pueden acceder a los discos o desplegar información

Perfomance : Se refiere a la capacidad que tiene el servidor para ejecutar sus aplicaciones, dependiendo de la memoria, procesador, acceso a disco, memoria RAM y acceso a la red.

Empresa : Unidad de organización dedicada a actividades industriales, mercantiles o de prestación de servicios con fines lucrativos.

Entidad : cualquier corporación, compañía, institución, etc., tomada como persona jurídica .

Hacker : Se le conoce así a las personas que se vuelven expertos en lenguajes de programación, pero aprovechan esto para introducirse ilícitamente a servidores

Backup : Se refiere a hacer una copia de información hacia un disco, una cinta o un CD, como medida preventiva en caso de que esta se pierda, y así poder recuperarla como si fuera la original.

Web site : Si en un servidor se tiene instalado un servidor de Web, se le conocerá como Web site. Este servidor permite tener paginas HTML para que puedan acceder a ellas desde cualquier otro equipo de computo, a través del puerto 80 (regularmente).

Scanneo : Este termino se puede usar para muchos casos, en este en particular se usa para definir una verificación a que puertos se encuentran disponibles en un servidor.