UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIA Y TECNOLOGÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA

TEMA:

"*Recomendaciones de seguridad para el servidor de la Universidad del Azuay*"

Trabajo de graduación previo a la obtención del Título de Ingeniero Electrónico

AUTOR:

Mateo Sebastián Encalada Guerrero

DIRECTOR:

Leopoldo Carlos Vásquez Rodríguez

CUENCA – ECUADOR

2011

DEDICATORIA

Dedico este trabajo de grado a mis padres y hermanos por su apoyo incondicional.

A G R A D E C I M I E N T O S

A la Facultad de Ciencia y Tecnología de la Universidad del Azuay.

Al Señor Licenciado Leopoldo Vásquez, por su invalorable aporte para la culminación de este trabajo.

Dejo constancia de mi agradecimiento a los Ingenieros: Leonel Pérez, Freddy Pesantez, Santiago Orellana, Juan Córdova, Hugo Torres, Francisco Vásquez y Edgar Pauta, por su valioso aporte.

Un especial agradecimiento a los Ingenieros Ernesto Pérez y Pablo Esquivel por su desinteresada colaboración en la elaboración de este trabajo.

Recomendaciones de seguridad para el servidor de la Universidad del Azuay

RESUMEN

Uno de los aspectos más importantes dentro de las comunicaciones y las redes, es la seguridad. Una buena seguridad brinda una mayor confiabilidad dentro de una institución. Para mejorar y analizar el estatus de seguridad del servidor, se pretende realizar el diseño de recomendaciones para la seguridad del mismo. Para esto se va a realizar un resumen del estatus de los servicios que brinda la universidad, para poder detectar los agujeros de seguridad y armar las recomendaciones para mejorarlos. Analizando los diferentes servicios que presta el servidor de la universidad, se encuentra algunas vulnerabilidades para evitar el robo de información por parte de un atacante.

PALABRAS CLAVE

agujeros, seguridad, redes, estatus, servicios.

Lcdo. Leopoldo Carlos Vásquez Rodríguez

Mateo Sebastián Encalada Guerrero

Safety Tips for the server of the University of Azuay

ABSTRACT

One of the most important aspects of communications and networks is security. A good security provides greater reliability within an institution. To improve and analyze the status of server security, is to make design recommendations for the safety of it. For this is going to make a summary of the status of services provided by the University, to identify security holes and assemble the recommendations for improvement. Analyzing the various services provided by the university server, there is some vulnerabilities to prevent theft of information by an attacker.

KEY WORDS

holes, security, networks, status, services.

Lcdo. Leopoldo Carlos Vásquez Rodríguez

Mateo Sebastián Encalada Guerrero

ÍNDICE DE CONTENIDOS

DEDICATORIA ..... **II**

A G R A D E C I M I E N T O S ..... **III**

RESUMEN ..... **IV**

ABSTRACT ..... **V**

INTRODUCCIÓN **1**

CAPITULO 1: SEGURIDAD LINUX

FUNDAMENTOS DE SEGURIDAD EN LINUX 2

Visión general de la seguridad en Linux *2*

Detección de intrusiones *5*

Seguridad Física *6*

Instalación: particiones y seguridad *6*

Administración de Linux *7*

SEGURIDAD DE LOS USUARIOS DE LINUX 9

Ataques a contraseña *10*

Código dañino *12*

SEGURIDAD DE LAS REDES LINUX 14

Sniffers y escuchas electrónicas *14*

Scanners *14*

Spoofing *17*

Secure Shell (ssh) *21*

SEGURIDAD LINUX EN INTERNET 21

Seguridad en FTP *21*

Seguridad en el correo *23*

Seguridad Telnet *25*

Seguridad de servidor Web *27*

Ataques de denegación de servicio *30*

Firewalls *30*

Logs y auditorías *33*
Detección de intrusiones *35*
Recuperación de desastres *36*
WEBMIN *37*
Backtrack *37*

CAPITULO 2: ESTATUS DEL SERVIDOR DE LA UDA

SEGURIDAD FÍSICA 40
CONTRASEÑAS 40
MYSQL 41
APACHE 41
MAILSCANNER (INCLUYEN CLAMMAV Y SPAMASSASSIN) 41
VSFTP 42
DOVECOT 3 42
SENDMAIL 42
DNS 43
SSH 43
FIREWALL 44
SCANNERS 44
NETENFORCER DE ALLOT 44
ASTARO SECURITY LINUX 44

CAPÍTULO 3: DISEÑO DE RECOMENDACIONES DE SEGURIDAD PARA EL SERVIDOR DE LA UDA

CONCLUSIÓN 47
REFERENCIAS BIBLIOGRÁFICAS 48
REFERENCIAS ELECTRÓNICAS 49
ANEXOS 50

INDICE DE FIGURAS

Figura 1.- Orden jerárquico del manejo de cuentas en Linux.....2
Figura 2.- Distintos permisos otorgables en Linux.....3
Figura 3.- Reglas de acceso a la red.....3
Figura 4.- Proceso de cifrado en internet.....4
Figura 5.- Registro y auditoría en Linux.....5
Figura 6.- Particionado del disco.....7
Figura 7.- Orden jerárquico del control de cuentas en Linux.....8
Figura 8.- Línea de permisos en Linux.....9
Figura 9.- Patrón de un scanner de sistema.....16
Figura 10.- Proceso de un scanner de red.....16
Figura 11.- Firewall basado en direccionador.....32
Figura 12.- Imagen del sistema de auditoría Backtrack 5 R1.....39

INDICE DE TABLAS

Tabla 1.- Comandos para la comunicación con un servidor SMTP 24

Mateo Sebastián Encalada Guerrero
Trabajo de Grado
Lcdo. Leopoldo Vásquez
Diciembre 2011

Recomendaciones de seguridad para el servidor de la Universidad del Azuay

INTRODUCCIÓN

Cuando un sistema es usado como un servidor en una red pública, se convierte en un objetivo para ataques. Por esta razón, y por razones de confiabilidad, es de suma importancia para el administrador fortalecer el sistema y bloquear servicios. La seguridad de servidores es tan importante como la seguridad de la red debido a que los servidores usualmente contienen una gran cantidad de información vital de la organización. Si un servidor está comprometido, todos sus contenidos pueden estar disponibles para que un “pirata” los manipule o robe a su gusto.

Muchos sistemas están expuestos a "agujeros" de seguridad que son explotados para acceder a archivos, obtener privilegios o realizar sabotaje. Estas vulnerabilidades ocurren por variadas razones, y miles de "puertas invisibles" son descubiertas (cada día) en sistemas operativos, aplicaciones de software, protocolos de red, browsers de Internet, correo electrónico y toda clase de servicios informáticos disponibles.

Los Sistemas operativos abiertos (como Unix y Linux) tienen agujeros más conocidos y controlados que aquellos que existen en sistemas operativos cerrados (como Windows). La importancia (y ventaja) del código abierto radica en que miles de usuarios analizan dicho código en busca de posibles bugs y ayudan a obtener soluciones en forma inmediata.

Constantemente encontramos en Internet avisos de nuevos descubrimientos de problemas de seguridad (y herramientas de Hacking que los explotan), por lo que hoy también se hace indispensable contar con productos que conocen esas debilidades, puedan diagnosticarlas y actualizar el programa afectado con el parche adecuado.

El propósito es encontrar los agujeros de seguridad en el servidor de la Universidad del Azuay, analizando el estatus de los diferentes servicios que presta. Se plantea diseñar recomendaciones que busquen eliminar dichos agujeros.

CAPITULO 1

SEGURIDAD LINUX

FUNDAMENTOS DE SEGURIDAD EN LINUX

Visión general de la seguridad en Linux

Componentes de la arquitectura de la seguridad de Linux

En la seguridad de Linux se manejan algunos componentes sumamente importantes para lograr un sistema seguro:

- Cuentas de usuario.
- Control de acceso discrecional.
- Control de acceso a la red.
- Cifrado.
- Conexión.
- Detección de intrusos.

Cuentas de usuario

En Linux todo gira alrededor de la cuenta “root”. La “root” es la cuenta administradora del sistema, la cual controla todo (incluyendo todas las demás cuentas).
Cada cuenta es una entidad independiente, con un nombre de usuario, una contraseña y unos derechos de accesos independientes.
En la figura 1 se muestra un ejemplo del orden jerárquico del manejo de cuentas en Linux.

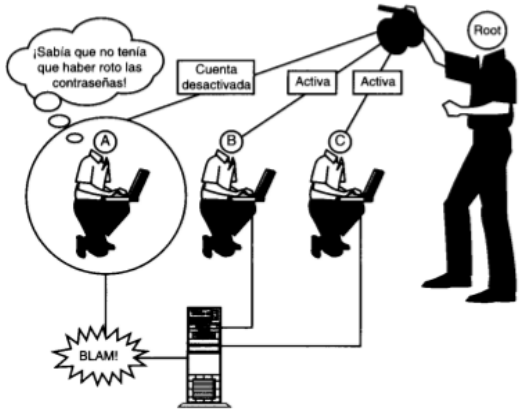

Figura 1.- Orden jerárquico del manejo de cuentas en Linux. Fuente: Linux Maximun Security

Para mantener el orden, cada usuario tiene su directorio y su espacio en el disco duro, evitando que la actividad de cada usuario afecte a la del sistema. El root controla el acceso y el lugar en el que cada usuario almacena sus archivos.

Control de acceso discrecional (DAC)

Es el control de los accesos a los archivos y directorios por los distintos usuarios. La figura 2 muestra los distintos permisos que pueden ser otorgados a los diferentes usuarios.

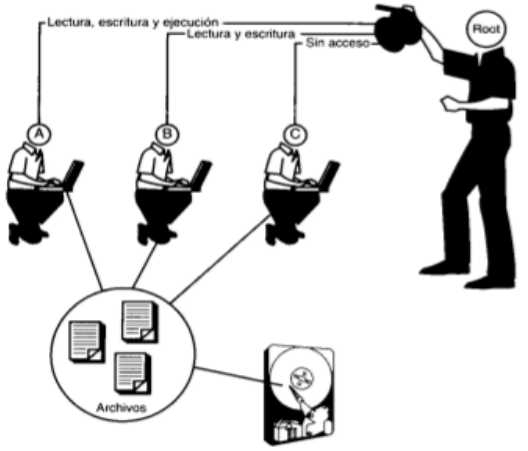

Figura 2.- Distintos permisos otorgables en Linux. Fuente: Linux Maximun Security.

Linux también permite crear grupos de usuarios, con permisos definidos para todo el grupo, sin necesidad de dar permisos a cada usuario.

Control de acceso a la red

Linux también proporciona control de acceso a redes. Como muestra la figura 3, es posible implementar reglas de acceso a la red.

Figura 3.- Reglas de acceso a la red. Fuente: Linux Maximun Security.

Cifrado

Además de la administración centralizada y del control de acceso a redes, Linux proporciona una gran variedad de mecanismos de cifrado. Cifrado es el proceso de mezclar los datos para que no puedan leerlos los que no tengan autorización para ello.

La figura 4 muestra el proceso de cifrado en la circulación de datos por internet.

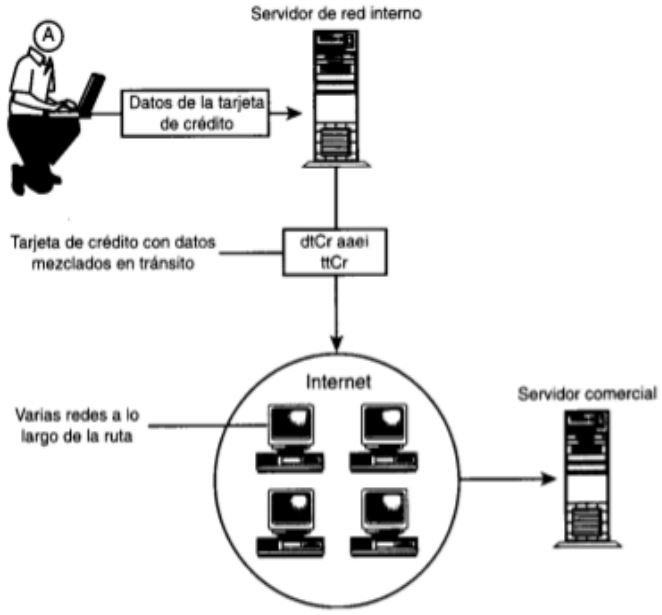

Figura 4.- Proceso de cifrado en internet. Fuente: Linux Maximun Security.

Registro, auditoría y control de red integrados

Linux puede registrar los movimientos de las personas que realizan algún ataque al sistema.

En la figura 5, se muestra como Linux detectará, marcará la hora y grabará las conexiones de red.

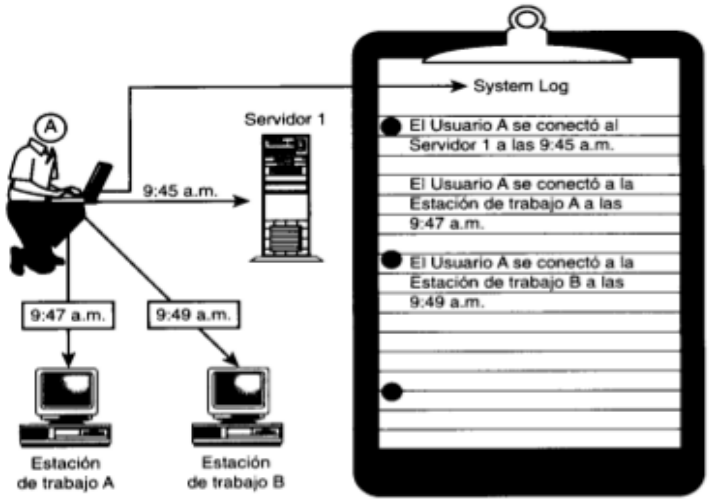

Figura 5.- Registro y auditoría en Linux. Fuente: Linux Maximun Security.

.

Linux graba registros a nivel de red, de host y de usuario:

- Registra todos los mensajes del sistema y del núcleo.
- Registra todas las conexiones de red, la dirección IP de la que parte cada una de ellas, su longitud y, en algunos casos, el nombre de usuario y sistema operativo de la persona que realiza el ataque.
- Registra los archivos que solicitan los usuarios remotos.
- Puede registrar que procesos se encuentran bajo el control de cualquier usuario.
- Puede registrar todos y cada uno de los comandos que ha emitido un usuario determinado.

Detección de intrusiones

La detección de intrusiones es una herramienta muy útil para evitar que los atacantes puedan ingresar al sistema u obtener la clave de root. Linux puede registrar los intentos de intrusión y que avise cuando se produzcan dichos ataques, acometa acciones predefinidas cuando los ataques cumplan unos criterios específicos, distribuya desinformación (engañando al atacante).

Seguridad Física

El primer objetivo dentro de una buena seguridad, es la Seguridad Física. Es más probable que un servidor sea atacado físicamente que informáticamente.

Aquí se proponen algunos tips para tener una buena seguridad física:

- El lugar en el que se encuentra el servidor debe ser seguro y las personas que tienen acceso a él deben ser confiables.
- Establecer centros de operaciones de red o NOC (área restringida donde se encuentran los servidores).
- Es recomendable elegir una topología de red en estrella (todas las estaciones de trabajo se conectan a un solo dispositivo de hardware, un *switch* o un *hub*).
- Si la red es grande, dividirla en segmentos, mejorando la gestión y la seguridad al limitar hasta dónde puede llegar un fallo de seguridad.
- Diseñe la red con tolerancia a fallos.
- Aísle el hardware.
- Aísle el cableado.
- Utilice hardware y software con posibilidad de cifrado en toda la LAN.
- Cambiar la configuración inicial de fábrica del hardware (contraseñas, cifrado, etc.).
- Establecer o cambiar contraseñas de BIOS (configuración del sistema) y consola (perfiles de usuario).
- Usar dispositivos de acceso biométrico.
- Instalar software o dispositivos de rastreo de marcado, si es que se utilizan Módems.
- Implantar dispositivos antirrobo como: Laptop Lockup, FlexLock-50, PHAZER, entre otros.

Instalación: particiones y seguridad

Las particiones son áreas del disco duro que se reservan para los sistemas de archivos.
En primer lugar, no se deben colocar los sistemas de archivo raíz y de usuario en la misma partición de Linux, sino los atacantes pueden explotar los programas SUID para acceder a áreas restringidas. Los archivos SUID siempre se ejecutan con permisos de propietario, independientemente de quien los ejecute. PE: si root tiene un programa SUID, este se ejecutara con privilegios de root y tienen posibilidad de accesar, modificar y sobrescribir archivos restringidos.

Aquí una imagen de cómo se debería particionar:

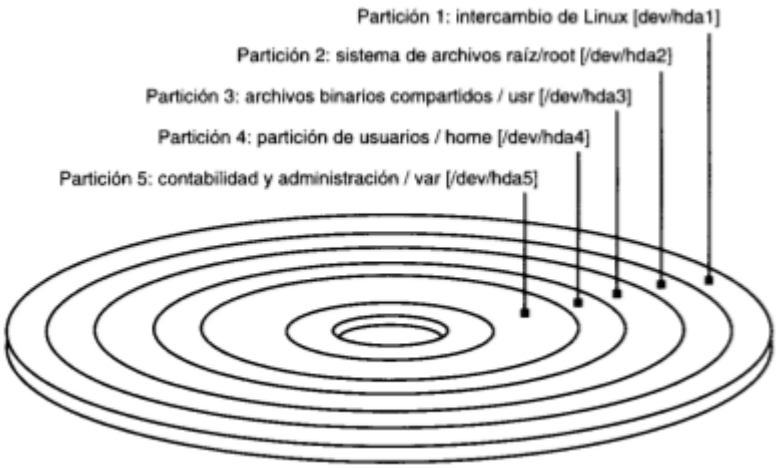

Figura 6.- Particionado del disco.

Ventajas de crear varias particiones:

- Sencilla gestión de copias de seguridad y actualizaciones.
- Arranque más rápido.
- Capacidad de control de montaje para cada sistema de archivos.
- Protección contra programas SUID renegados.

Por problemas de seguridad, no se recomienda la instalación de los servicios que vienen por defecto o que son opcionales. A veces se instalan servicios que no van a ser utilizados y posteriormente desatendidos, lo que generara un enorme hueco en la seguridad del sistema.

Administración de Linux

Todo el poder administrativo se otorga al root. Este controla a los usuarios, grupos y a los archivos en un orden jerárquico. La siguiente figura muestra dicho orden.

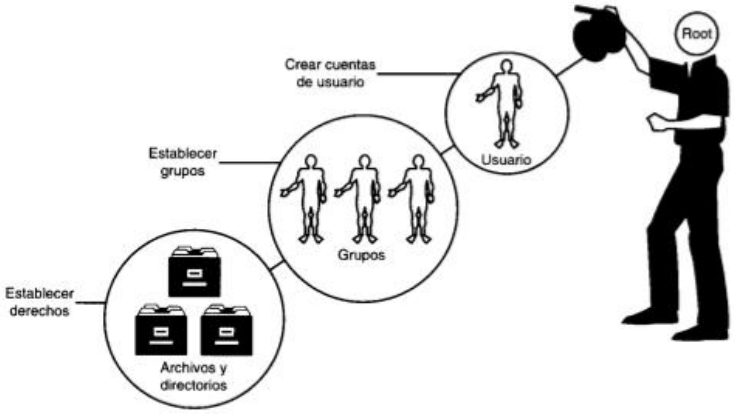

Figura 7.- Orden jerárquico del control de cuentas en Linux. Fuente: Linux Maximun Security.

Su propia cuenta

Es la primera cuenta que se va a crear, y porque crearla?, si ya se tiene el root? La cuenta root no debe ser utilizada para fines personales, ya que tiene poder absoluto, y en ciertos casos, se pueden cometer errores involuntarios, lo que sería un desastre para el sistema.

Crear y administrar cuentas

Uno de los aspectos más importantes para la seguridad en la creación y administración de cuentas, es si van a tener autorización para iniciar una sesión o para acceder a los servicios, lo que es llamado la Política de cuentas. Solo si es imprescindible, se consideran estos permisos a alguna de las cuentas.

Aquí algunas medidas para reducir los riesgos en caso de existir estos permisos:

- Dedique una máquina exclusivamente para el acceso a la shell.
- Restrinja dicha máquina solo para el uso de la shell.
- Elimine de ella todos los servicios de red innecesarios.

Realizar tareas administrativas con *su*

Con frecuencia se necesitará utilizar la potencia de root para administrar el sistema, para ello va a utilizar *su*.

***su*, el usuario sustituto** permite ejecutar una shell ajena.
A veces, se necesita delegar responsabilidades limitadas a otros usuarios, para ello se utiliza *sudo*.
Sudo: permite a los usuarios elegidos ejecutar determinados comandos como si fueran root.

Control de acceso

Es una técnica de manejo de accesos a los recursos del sistema.

Permisos y propiedad

El acceso de los usuarios a los distintos **archivos y directorios** se maneja mediante la concesión de permisos. Hay 3 tipos de permisos:

- De lectura (r).
- De escritura (w).
- De ejecución (x).

En la figura 8 se explica qué significa una línea de permisos en el comando *ls –l*;

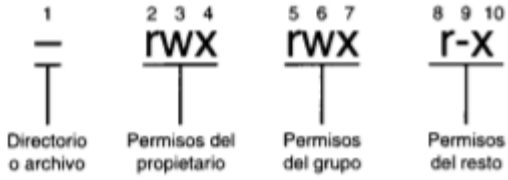

Figura 8.- Línea de permisos en Linux. Fuente: Linux Maximun Security.

SEGURIDAD DE LOS USUARIOS DE LINUX

Uno de los aspectos más importantes en la seguridad de Linux es la **seguridad de las contraseñas**. Por un lado, se aplican herramientas para reforzar las contraseñas, y por otro, es necesario educar a los usuarios sobre políticas de contraseñas básicas.

Ataques a contraseña

Un ataque a contraseña es cualquier acción dirigida a romper, descifrar o borrar contraseñas o a sortear los mecanismos de seguridad de las mismas.

Generación y almacenamiento de contraseñas en Linux

Las contraseñas de Linux se almacenaban en el directorio /etc/passwd y se creaban utilizando un avanzado algoritmo de cifrado llamado *Data Encryption Standard* o DES.

Data Encryption Standard (DES)

DES es un cifrado de bloque, que trabaja sobre bloques de datos de un determinado tamaño (64 bits). Los bloques de datos que superan este tamaño se dividen en fragmentos de 64 bits. Las porciones restantes inferiores a 64 bits se rellenan, quiere decir, que DES añade bits sin significado a partes más pequeñas para conseguir un bloque completo de 64 bits.

A partir de aquí, DES efectúa tres operaciones importantes:

- **Permutación**: los bits de datos se desplazan a otras posiciones en una tabla. Por ejemplo: se tiene la frase “EL COCHE ROJO”, permutando saldría “ECR LOO CJ HO E”.
- DES produce un **bloque de entrada**: este bloque se reordena mediante complicadas operaciones matemáticas para crear un bloque de pre-salida.
- **Bloque de pre-salida**: al bloque se le aplica otra permutación más y el resultado final es el texto codificado.

Si un sistema almacena las contraseñas de esta manera, es conveniente instalar el *shadowing* de contraseñas manualmente. A veces, para que un atacante encuentre la clave correcta, solo puede concatenar /etc/passwd con un archivo y utilizar las claves cifradas para llevar a cabo un **ataque a diccionario**.

Ataques a diccionario

Las contraseñas de Linux codificadas con DES pueden romperse rápidamente, por las siguientes 2 razones:

- Elección de contraseñas débiles.
- Las contraseñas de Linux son cortas.

En los ataques a diccionario, los atacantes toman diccionarios (grandes listas de palabras) y los codifican utilizando DES. Con el paso del tiempo, utilizando herramientas de ruptura de alta velocidad, los atacantes pueden codificar cada palabra del diccionario de 4096 formas diferentes. Cada vez que una herramienta de ruptura obtiene dicho texto codificado, lo compara con las contraseñas de /etc/passwd. Cuando encuentra una coincidencia, comunica al agresor que se ha roto una contraseña.

Durante años, los intrusos se han dirigido a /etc/passwd porque era donde se almacenaba las contraseñas de los usuarios. En consecuencia, los especialistas en seguridad de UNIX se vieron forzados a reconsiderar la seguridad de las contraseñas. Necesitaban una forma de que /etc/passwd fuera legible, al mismo tiempo que oscurecían las contraseñas cifradas, entonces crearon el *shadowing* de contraseñas.

Shadowing de contraseñas y la suite shadow

El shadowing de contraseñas es una técnica mediante la que el archivo /etc/passwd sigue siendo legible pero ya no contiene las contraseñas. En su lugar, se almacenan en /etc/shadow.

A diferencia del /etc/passwd, el /etc/shadow implementa 2 nuevos conceptos:

- Vencimiento de la contraseña.
- Bloqueo automático de cuenta: cuando los usuarios no hacen caso al vencimiento de contraseña, se les bloquea la cuenta.

Ataques con shadowing

Ahora que las contraseñas están ocultas en la suite shadow, los atacantes han volcado su interés hacia /etc/shadow. La única diferencia es que /etc/shadow es más difícil de alcanzar. Desafortunadamente la seguridad de la /etc/shadow depende mucho de la seguridad del sistema, ya que muchas otras aplicaciones tienen agujeros que permiten a los atacantes leer o escribir en /etc/shadow. La única forma de proteger las contraseñas ocultas es permanecer alerta y mantener el sistema actualizado.

A más de la instalación del shadowing de contraseñas, es necesario ampliar el alcance de la seguridad tradicional de las contraseñas, hasta:

- Elección humana de contraseñas y seguridad del sistema.
- Comprobación proactiva de contraseñas.
- Seguridad auxiliar de las contraseñas.

Elección humana de contraseñas y seguridad del sistema

Es muy importante que todos los usuarios de un sistema elijan una contraseña que no se adivine con facilidad. La seguridad de cada uno de los usuarios es importante para la seguridad de todo el sistema. La mejor solución al problema de poseer contraseñas fáciles de adivinar es evitar que se introduzcan al sistema una primera vez. Si es que estas contraseñas ya se encuentran dentro del sistema, se podría decir que desde el comienzo ya hubo un agujero de seguridad en el sistema. Si un programa que cambia las contraseñas de los usuarios, comprueba su seguridad y la posibilidad de que las averigüen antes de que la contraseña haya sido asociada a la cuenta del usuario, nunca habrá existido ningún fallo en la seguridad. A esta técnica se la llama **comprobación proactiva de contraseñas.**

Comprobación proactiva de contraseñas

Funciona eliminando las contraseñas débiles antes de que se manden a la base de datos de contraseñas. El proceso funciona de la siguiente manera:

- El usuario crea la contraseña.
- Ésta se compara con una lista de palabras y una serie de reglas.
- Si la contraseña no cumple los requisitos (posee muchas coincidencias o es sencilla), se obliga al usuario a elegir otra.

Código dañino

El código dañino es un código no autorizado que realiza funciones que el usuario no conoce.

Existen 2 tipos importantes de código dañino:

- Troyanos

- Virus

Troyanos

Los troyanos son cualquier programa (normalmente legal) que ha modificado algún programador malicioso, insertando código adicional que va a ejecutar una función oculta no autorizada.

Virus

Los virus informáticos se encuentran en 2 categorías:

- Programas diseñados para infectar, modificar o sobrescribir el sector de arranque o el registro de inicio maestro.
- Programas diseñados para adjuntar código dañino a los archivos del objetivo.

Detección de código dañino

El método más fiable de detectar código dañino es la reconciliación de objetos que consiste en comparar el estado actual del sistema con una “instantánea” del sistema operativo hecho inmediatamente después de la instalación.

Existen varios métodos para realizar la reconciliación de objetos:

- Generar una lista de comprobación: se examina esta lista para ver si se ha suscitado algún cambio en: la última fecha en que se han modificado, su fecha de creación o su tamaño.
- Sumas de comprobación básicas: son valores numéricos que se componen de la suma de los bits de un archivo almacenados por el cliente y el servidor y que van a ser comparados después de una transmisión para saber si ha existido alguna clase de manipulación y si los datos son peligrosos.
- MD5: es un algoritmo de huella digital, que toma como entrada un mensaje de longitud arbitraria y crea como salida una “huella digital” de 128 bits de la entrada.

SEGURIDAD DE LAS REDES LINUX

Sniffers y escuchas electrónicas

Los sniffers son dispositivos de monitorización ocultos que recogen la información de la red. Pueden capturar nombres de usuario y contraseñas o grabar todo el tráfico de la interfaz de red.

Funcionamiento de los *sniffers*

Por default, las estaciones de trabajo escuchan y responden solamente a los paquetes que van dirigidos a ellas pero es posible modelar el software que lanza la interfaz de red de una estación de trabajo a modo promiscuo, entonces, ésta puede monitorizar y capturar todo el tráfico de red y los paquetes que pasen por ella, independientemente del destino que tengan.

Riesgos de los sniffers

Los sniffers presentan una alto nivel de riesgo porque pueden capturar contraseñas, información confidencial u obtener acceso por la fuerza a un sistema.

Defenderse contra ataques de sniffers

Para detectar un sniffer hay que averiguar si alguna de las interfaces de la red se encuentra en modo promiscuo, para lo que pueden utilizarse las siguientes herramientas:

- ifconfig
- ifstatus

Pero la medida preventiva más eficaz contra los sniffers desde el principio, cuando se establezca la red, es el cifrado.

Scanners

Un scanner es una herramienta de seguridad que detecta los puntos vulnerables del sistema.

Hay 2 categorías de scanners:

- Scanners de sistema.
- Scanners de red.

Scanner de sistema

Los scanners de sistema rastrean hosts locales en busca de puntos vulnerables de la seguridad que aparecen a causa de los descuidos y negligencias, y los problemas de configuración que olvidan los usuarios.

Scanner de red

Los scanners de red prueban hosts sobre conexiones de red, de forma similar a como lo haría un intruso. Examinan los servicios y puertos disponibles en busca de debilidades conocidas que pueden explotar los atacantes remotos.

Funcionamiento de los scanners

Es un proceso lógico que tiene el siguiente patrón:

- Cargan un conjunto de reglas o una serie de ataques.
- Prueban el objetivo con estos parámetros.
- Informan de los resultados.

En la siguiente figura se muestra el patrón que siguen los scanners de sistema.

Figura 9.- Patrón de un scanner de sistema. Fuente: Linux Maximun Security.

La siguiente figura muestra el proceso de los scanners de red:

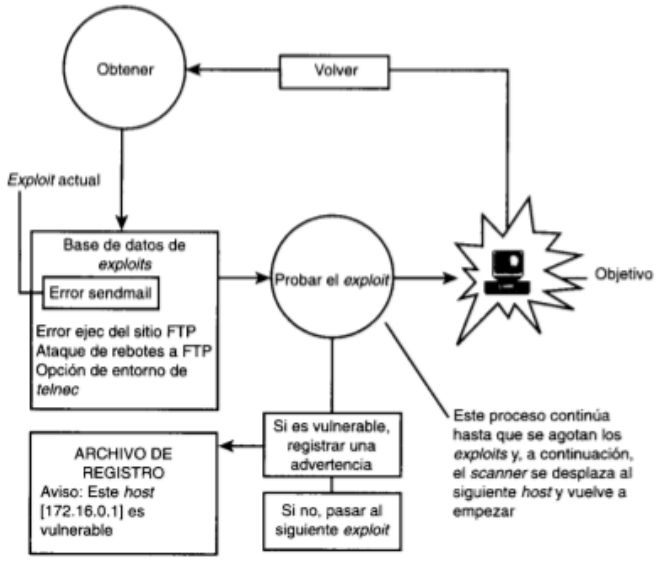

Figura 10.- Proceso de un scanner de red. Fuente: Linux Maximun Security

Spoofing

El spoofing se produce cuando los atacantes autentican una máquina con otra mediante la falsificación de paquetes de un host en el que se confía.

Hay varias técnicas de spoofing, entre las más importantes se tienen:

- Spoofing de IP.
- Spoofing de ARP.
- Spoofing de DNS.

Spoofing de TCP e IP

Las diferentes herramientas que controlan el acceso a la red, como los empaquetadores de TCP o los firewalls, confían en la fuente o en la dirección IP como identificador. La utilización de la dirección de origen para la autentificación representa un serio agujero en la seguridad de TCP/IP.

Las partes importantes de la cabecera de TCP son un número de puerto de origen, un número de puerto de destino, un número de secuencia, un número de confirmación y algunas marcas. Los números de puerto identifican los circuitos virtuales implicados, los números de secuencia y de confirmación garantizan que los datos se reciben en el orden correcto y las marcas afectan al estado del circuito virtual. Una cabecera de IP consta de identificadores de los hosts de origen y de destino; dichos identificadores son números de 32 bits que indican exclusivamente un host y una red. Por tanto, la dirección de origen es un identificador exclusivo no fiable.

4.2BSD proporciona un servidor de ejecución remota que escucha las solicitudes de las conexiones TCP. Cuando dichas solicitudes llegan a una máquina, el servidor comprueba que el host del que parte dicha solicitud es “de confianza” comparando el Id. del host de origen de la cabecera IP con una lista de equipos en los que se confía. Si el host de origen es correcto, el servidor lee los Id. de usuario y comandos que se van a ejecutar desde el circuito virtual que proporciona TCP. El punto débil de este esquema es que el propio host de origen rellena el Id. de la IP del host de origen y no hay provisión en 4.2BSD ni en TCP/IP para descubrir el verdadero origen de un paquete.

En resumen, solo se necesitaría falsificar la dirección de origen para poder autenticarse. Pero

no es así de sencillo. Hay otros factores que complican el spoofing, entre ellos, los “números de secuencia”.

El número de secuencia se utiliza para confirmar la recepción de los datos. Al principio de cualquier conexión TCP, el cliente envía un paquete TCP con un número de secuencia inicial. El servidor en el otro extremo de la conexión devuelve un paquete TCP con su propio número de secuencia inicial y una confirmación: el número de la secuencia inicial del paquete del cliente más uno. Cuando el sistema cliente recibe este paquete, debe devolver su propia confirmación: el número de secuencia inicial del servidor más uno.

Entonces el atacante tiene 2 problemas: falsificar la dirección de origen y mantener un diálogo de secuencias con el destino. Este último es más complicado, ya que los números de secuencia no son arbitrarios. 4.2BSD mantiene un número global de secuencia inicial, que se incrementa en 128 cada segundo y en 64 cada vez que se inicia la conexión; cada nueva conexión comienza con este número. Cuando se envía un paquete SYN con un origen falsificado desde un host, el host destinatario enviará la respuesta al supuesto host de origen, no al que está realizando la falsificación. Este debe descubrir o averiguar el número de secuencia de dicho paquete perdido para confirmarlo y poner el puerto TCP de destino en estado ESTABLISHED. Si el atacante averigua el número de secuencia, puede establecer una sesión válida, y su máquina estaría conectada al destino como host de confianza y ahí el atacante podría iniciar sesión.

Servicios vulnerables al Spoofing de IP

- RPC (Remote Procedure Call).
- Los servicios que utilizan autenticación de direcciones IP.
- Servicios R.
- X Windows (interacción gráfica en red para sistemas Unix).

Pero el Spoofing no necesariamente apunta a causar problemas de autenticación e inicio de sesión sino también a causar ataques de “Bucle”. Los ataques de “Bucle” hacen que éstos se envíen entre sí mensajes de error continuamente.

Evitar ataques de Spoofing de IP

El secreto de la defensa contra el spoofing es no utilizar la dirección de origen para la

autenticación. Actualmente no hay ninguna razón para realizar dicha autenticación, para eso existen soluciones criptográficas como SSH. Los números de secuencia válidos no son un sustituto de la autenticación criptográfica. Alguien que pueda observar los mensajes iniciales de una conexión puede determinar el estado de su número de secuencia e iniciar ataques de averiguación de números de secuencia mediante la imitación de dicha conexión.

Si se desea no instituir la autenticación criptográfica en todo el sistema, se recomienda lo siguiente:

- Configurar el *router* para que rechace paquetes de la Red que se originan desde una dirección local. También tendrá que configurar el *firewall* para denegar el acceso a direcciones internas.
- Si se permiten conexiones externas desde hosts confiables, active las sesiones de cifrado en el router. Con ello se evita que los atacantes capturen el tráfico de la red para realizar muestreos.

También se puede detectar el Spoofing a través de procedimientos de registro. Se trata de realizar una comparación de conexiones entre host confiables que tienen una sesión en directo, ambos mostrarán procesos que indican que la sesión está en ejecución. Si uno de ellos no lo hace, podría estar en proceso un ataque de Spoofing.

Spoofing de ARP

ARP (Address Resolution Protocol) resuelve las direcciones IP en direcciones MAC. Cuando un host desea una sesión, envía una difusión de ARP que lleva la dirección IP del objetivo deseado. El sistema proporciona un caché de ARP para que las máquinas se conecten a hosts conocidos sin necesidad de esta difusión. Es esta cache el objetivo de los atacantes (La caché de ARP contiene información de asignación de hardware a IP).

El spoofing de ARP también utiliza direcciones para la autenticación. La diferencia es que ARP confía en las direcciones de red. En el spoofing de ARP, el objetivo del atacante es conservar su MAC, al mismo tiempo que la dirección IP de un host en el que se confía. Para ello, el atacante envía información falsa de asignación al objetivo y a la caché. Entonces, los paquetes del objetivo se encaminan a la dirección de hardware del atacante. El objetivo cree que la máquina del atacante es realmente el host en el que se confía.

Una de las limitaciones de los ataques de spoofing de ARP, es que el atacante tiene una pequeña oportunidad para realizar el ataque porque las entradas de la caché expiran rápidamente (más o menos cada 5 min.).

Defenderse contra los ataques de Spoofing de ARP

La forma más eficaz de defenderse contra los ataques de spoofing ARP, es escribir las asignaciones de direcciones en piedra. Una manera de realizar esto es crear entradas estáticas en la caché de ARP, con lo que no caducan cada pocos minutos. Pero no hay que olvidar que requiere que se actualice la caché manualmente cada vez que cambia una dirección de hardware.

Spoofing de DNS

En el Spoofing de DNS, el intruso pone en peligro el servidor de DNS y modifica las tablas de direcciones IP del nombre del host. Entonces, cuando un cliente solicita una búsqueda, recibe una dirección falsa. Esta dirección es la dirección IP de una máquina que está bajo el control total del intruso.

Este tipo de ataques no son muy comunes. En una DNS simulada, es posible que los intrusos emulen BIND para proporcionar nombres incorrectos. Algunos sistemas y programas dependen de esta información para la autenticación, por lo que es posible obtener acceso no autorizado.

Detectar y defenderse contra spoofing de DNS

El spoofing de DNS es fácil de detectar. Si sospecha que alguno de los servidores DNS está bajo ataque, la solución es sondear los demás servidores de DNS autorizados de la red. Si éstos generan resultados que variarán de los que proporciona el servidor DNS sospechoso, está ante un ataque de spoofing de DNS. A veces es posible que este sondeo no sea suficiente para detectar un servidor DNS afectado. Es posible que se hayan pasado tablas de direcciones de host falsas a otros servidores DNS de la red. Si se observan anomalías en la resolución de los nombres, se puede utilizar un script llamado DOC. DOC es un programa que diagnostica dominios que no se comportan correctamente mediante el envío de consultas a los servidores de nombre apropiados del dominio y la realización de una serie de análisis en la salida de estas consultas.

Secure Shell (ssh)

SSH es un sistema de inicio de sesión seguro. Es un programa para conectarse a otro equipo a través de una red, para ejecutar comandos en una máquina remota y para mover archivos de una máquina a otra. Ssh admite varios algoritmos como Blowfish, Triple DES, IDEA, o RSA. La arquitectura de ssh es tal que al protocolo básico le da igual el algoritmo que se utilice. Ssh no modifica las rutinas.

SEGURIDAD LINUX EN INTERNET

Seguridad en FTP

El protocolo de transferencia de archivos o FTP es el método estándar de transferencia de archivos de un sistema a otro.

Los objetivos de FTP son:

- Promover la compartición de archivos.
- Fomentar el uso de computadoras remotas.
- Proteger a los usuarios de la diferencias en los sistemas de almacenamiento de archivos entre los hosts.
- Transferir datos de una manera eficaz y fiable.

Las deficiencias en aspectos de seguridad de FTP son:

- FTP utiliza la autenticación estándar de nombres de usuario/contraseñas, por lo tanto, el servidor no puede determinar de manera fidedigna si un determinado usuario es realmente quien afirma ser.
- De forma predeterminada, las contraseñas se transmiten en texto sin formato, donde el atacante puede utilizar sniffers para capturar contraseñas.
- Las sesiones de FTP no están cifradas. No hay privacidad.

Vulnerabilidades de FTP

Las vulnerabilidades históricas de FTP son:

- Ataques de rebote a FTP.
- Permisos de archivos erróneos.
- El error SITE EXEC (ya solucionado).

Ataques de rebote a FTP

Cuando un intruso quiere acceder a una máquina objetivo, que tiene restringida su dirección IP, utiliza otra máquina (una intermediaria) para acceder al objetivo. Para ello, el intruso escribe un archivo en el directorio FTP del intermediario que contiene comandos para conectarse con el objetivo y recuperar archivos. Como la conexión proviene de la dirección del intermediario, el objetivo acepta la conexión solicitada y envía el archivo específico.

Permisos erróneos

Antiguamente, los atacantes habían conseguido acceso a root aprovechándose de los permisos erróneos de archivos y directorios de sus objetivos. Si va a ejecutar un FTP anónimo, compruebe los permisos FTP de los siguientes archivos para cerrar cualquier tipo de agujero en este sentido: [ftp-home]ftp, [ftp-home]ftp/bin, [ftp-home]ftp/bin/ls, [ftp-home]ftp/etc y [ftp-home]ftp/etc/passwd.

El error SITE EXEC

El error SITE EXEC permite que usuarios individuales remotos obtengan una shell al iniciar una sesión telnet con el puerto 21.

Características de seguridad de FTP

FTP ofrece características de seguridad que incluyen el control de acceso a la red basado en el host y en el usuario. Estas características se implementan utilizando 3 archivos:

- /etc/ftpusers: archivo de acceso restringido a los usuarios. Cualquier usuario dentro de este archivo tiene denegado el acceso al *login* de FTP.
- /etc/ftphosts: archivo de acceso de usuarios/hosts individuales de ftpd. Se utiliza para conceder o denegar el acceso a determinadas cuentas de varios hosts.
- /etc/ftpaccess: archivo de configuración ftpd. Se controla la manera de funcionar de ftpd.

Es posible que estas medidas de seguridad de FTP sean suficientes en redes cerradas de

pequeño tamaño sin conexión a internet, pero en entornos de red con mayor alcance (sobre todo los que cuentan con conexión a internet), es conveniente utilizar SSLftp.

SSLftp

SSLftp es un cliente y servidor FTP con SSL activo. SSL es Secure Sockets Layer, un protocolo y una API (Application Programming Interface) de tres partes que emplea la autenticación y el cifrado RSA y DES, así como la verificación adicional de la integridad de la sesión MD5.

Seguridad en el correo

Clientes y servidores SMTP

El protocolo de transporte de e-mail más utilizado es SMTP (Protocolo simple de transferencia de correo). El funcionamiento de los servidores SMTP es muy sencillo, aceptan un mensaje entrante y comprueban las direcciones del mensaje, si son direcciones locales, almacenan el mensaje para recuperarlo pero si son remotas, envían el mensaje. Los mensajes pasan a través de varios gateways SMTP antes de llegar a su destino final. En cada parada, los servidores SMTP evalúan el mensaje y lo envían. Pero si el servidor SMTP encuentra un mensaje que no se puede enviar, SMTP devolverá un mensaje de error al remitente que explica el problema.

No siempre es necesario comunicarse con un servidor SMTP utilizando un cliente especial de e-mail, sino es posible interactuar con él directamente utilizando inglés casi normal mediante una serie de comandos, a través de una sesión de telnet con el puerto 25.

La tabla 1 describe los comandos que se pueden utilizar para comunicarse directamente con un servidor SMTP.

Tabla 1.- Comandos para la comunicación con un servidor SMTP

Comando	Propósito
DATA	Este comando se utiliza para especificar que las líneas de texto siguientes son el cuerpo de un mensaje de correo electrónico. El final del mensaje se indica mediante el envío de una línea en la que haya un solo punto.
EXPAND	Este comando se utiliza para expandir un nombre de usuario a una dirección de correo plenamente calificada.
HELO (HELLO)	Este comando se utiliza para iniciar una sesión de SMTP e intercambiar datos de identificación.
HELP	Este comando se utiliza para obtener ayuda sobre SMTP.
MAIL	Este comando se utiliza para iniciar una transacción de e-mail.
QUIT	Este comando se utiliza para finalizar la sesión actual y cerrar la conexión.
RCPT (RECIPIENT)	Este comando se utiliza para especificar un destinatario.
RESET	Este comando se utiliza para detener la operación actual.
SEND	Este comando se utiliza para iniciar el envío.
VERIFY	Este comando se utiliza para verificar un nombre de usuario.

Uno de los problemas al utilizar servidores SMTP, es que confían en todo el mundo. Los usuarios pueden especificar la dirección de retorno que deseen y los servidores SMTP procesarán el correo utilizando esta dirección falsa. Además, al interactuar directamente con servidores SMTP, los intrusos tienen la ventaja de obtener anonimato de su e-mail si eligen un servidor que ya está en peligro.

Los servidores SMTP constituyen un reto para la seguridad y exigen que se centre en dos tareas distintas:

- Proteger a los servidores de intrusos.
- Proteger a los servidores SMTP de un uso incorrecto, como el spam.

Principios básicos de la seguridad de Sendmail

Sendmail es un agente de transporte de correo, complejo, eficaz y difícil de configurar. El objetivo de los intrusos es sendmail, porque es un servicio público disponible, suele ejecutarse

como root y puede estar mal configurado.

Protección de los servicios de Sendmail

La mejor defensa contra los ataques de Sendmail es mantenerse actualizado todo el tiempo. Sin embargo hay algunos pasos para proteger los servicios de Sendmail.

Listas negras en tiempo real

Existe una lista de las personas que envían correo basura y en la cual Sendmail realiza consultas dinámicas para decidir si aceptar o no correo de un dominio determinado. Esta lista se llama *Realtime Blackhole List* (RBL) y es pública. Dicha lista se mantiene actualizada gracias a la contribución de administradores de todo el mundo. Su funcionamiento es sencillo. RBL es un servidor DNS modificado que responde a las consultas de nombres de forma exclusiva.

Si se configura MTA (Mail Transfer Agent) para evitar las transmisiones abiertas y el pirateo de cuentas, se protege la red, el servidor y a los usuarios. Sendmail ofrece un servicio SMTP de gran potencia y una excelente compatibilidad con las utilidades existentes de Linux/UNIX.

Seguridad Telnet

Es un protocolo de red que sirve para acceder mediante una red a otra máquina para manejarla remotamente. Telnet, o una variante de éste, es una necesidad real. Existen muchas tareas que se pueden realizar fácilmente con telnet y que, de otra manera, serían muy difíciles. Pero esto no quiere decir que se deba brindar acceso a todos los usuarios, todo lo contrario, no permita al público el acceso a telnet o shell.

Telnet sólo sirve para acceder en modo terminal y fue una herramienta muy útil para arreglar fallos a distancia, sin necesidad de estar físicamente en el mismo sitio que la máquina que los tenía. También se usaba para consultar datos a distancia, como datos personales en máquinas accesibles por red, información bibliográfica, etc. En general **telnet** se ha utilizado (y aún hoy se puede utilizar en su variante SSH) para abrir una sesión con una máquina UNIX, de modo que múltiples usuarios con cuenta en la máquina, se conectan, abren sesión y pueden trabajar utilizando esa máquina. Es una forma muy usual de trabajar con sistemas UNIX.

Problemas de seguridad y ssh

Su mayor problema es de seguridad, ya que todos los nombres de usuario y contraseñas necesarias para entrar en las máquinas viajan por la red como *texto plano* (cadenas de texto sin cifrar). Esto facilita que cualquiera que espíe el tráfico de la red pueda obtener los nombres de usuario y contraseñas, y así acceder él también a todas esas máquinas. Por esta razón dejó de usarse, casi totalmente, hace unos años, cuando apareció y se popularizó el SSH, que puede describirse como una versión cifrada de **telnet** (actualmente se puede cifrar toda la comunicación del protocolo durante el establecimiento de sesión).

Hoy en día este protocolo también se usa para acceder a los BBS (software que permite a los usuarios conectarse con el sistema, y utilizando un programa terminal o telnet, realizar funciones como descargas, mensajería, juegos en línea, etc.), que inicialmente eran accesibles únicamente con un módem a través de la línea telefónica.

Seguridad

Hay tres razones principales por las que el telnet no se recomienda para los sistemas modernos desde el punto de vista de la seguridad:

- Los dominios de uso general del telnet tienen varias vulnerabilidades descubiertas sobre los años, y varias más que podrían aún existir.
- Telnet, por defecto, no cifra ninguno de los datos enviados sobre la conexión (contraseñas inclusive), así que es fácil interferir y grabar las comunicaciones, y utilizar la contraseña más adelante para propósitos maliciosos.
- Telnet carece de un esquema de autentificación que permita asegurar que la comunicación esté siendo realizada entre los dos anfitriones deseados, y no interceptada entre ellos.

No se debe utilizar telnet en ambientes donde es importante la seguridad, por ejemplo en el Internet público. Las sesiones de telnet no son cifradas. Esto significa que cualquiera que tiene acceso a cualquier router, switch, o gateway localizado en la red entre los dos anfitriones donde se está utilizando telnet, puede interceptar los paquetes de telnet que pasan cerca y obtener fácilmente la información de la conexión y de la contraseña con cualesquiera de varias utilidades comunes como *tcpdump* y *Wireshark*.

Seguridad de servidor Web

Linux ofrece una característica en particular que le ha permitido introducirse en el mercado empresarial: puede transformar PC's baratos en servidores web con todas las garantías. Este tema se centra en asegurar hosts web.

Eliminación de servicios no esenciales

La primera decisión crucial en asegurar su host web es determinar qué tipo de host está construyendo. Los tres tipos más comunes son:

- Host web de intranet. Host de área local sin conexión a internet.
- Host web privado. Host con conexión a internet que proporciona servicios a usuarios privados.
- Host web público. Host para usuarios públicos con conexión a internet.

Normalmente las distribuciones Linux por defecto incluyen servicios, muchas de las veces, innecesarios, entre los cuales están:

- FTP (File Transfer Protocol).
- finger.
- NFS (Network File System).
- Servicios RPC (Remote Procedure Call) (SOAP, XML-RPC).
- Servicios R.
- SMB (Server Message Block).

FTP

FTP es el método para transferir archivos de un sistema a otro. En intranet y en los host web privados, puede proporcionar ftp como una manera segura de distribuir archivos o como una vía alternativa de recuperar información que de otro modo está disponible vía HTTP.

Para servidores web públicos debería dar FTP público, pero esto supone algunos riesgos de seguridad:

- Si los atacantes comprometen el servidor FTP, pueden tener acceso a los recursos del

host restantes.

- Los atacantes pueden usar el FTP externo para burlar el firewall.
- Los atacantes pueden realizar ataques de saturación de disco en servidores FTP con directorios que tienen permiso de escritura.
- Usuarios ajenos pueden utilizar el servidor FTP para almacenar contrabando.

finger

fingerd (el servidor finger) presenta información personal de los usuarios que se especifiquen, incluido el nombre de usuario, nombre real, shell y el directorio.
finger no es esencial y puede ser utilizado por los atacantes como un centro de información creando listas de usuarios, estableciendo así, otras posibles vías de ataque.

NFS

El Sistema de Archivos de Red proporciona acceso a directorios y archivos distribuidos y permite a usuarios de hosts remotos montar su sistema de archivos desde lejos. En la máquina del usuario remoto, sus sistemas de archivos exportados parecen y actúan como si fueran locales. En redes internas, puede utilizar NFS para distribuir una jerarquía de directorios central (brindar herramientas a un grupo de trabajo) o para distribuir directorios matrices de usuario (los usuarios tienen acceso a sus archivos incluso cuando hacen *log in* a diferentes máquinas).

Si se requiere utilizar NFS en un servidor web interno, haga lo siguiente:

- Considere crear una división separada para los sistemas de archivos exportables y active *nosuid*.
- Exportar sistemas de archivos de solo lectura.
- Limite el acceso a portmapper a los hosts de confianza.
- NUNCA exportar el sistema de archivos raíz.

Solo si es absolutamente necesario, puede ejecutar NFS en un servidor público. Para un mejor y más sencillo manejo de este tipo de servicios, se puede utilizar SAMBA.

SAMBA

Samba es un software que le permite al ordenador compartir archivos e impresoras con otras

computadoras en una red local. Utiliza para ello un protocolo conocido como SMB/CIFS compatible con Linux, Windows y OS X.

Servicios RPC

- **ruserd:** proporciona la misma información que finger.
- **rstatd:** proporciona información de las estadísticas de la CPU, memoria virtual, tiempo de conexión a la red y disco duro.
- **rwalld (el servidor rwall):** permite a los usuarios remotos enviar mensajes a todos los usuarios de la red.

Servicios R

- **rshd (el servidor de *shell* remota):** permite la ejecución remota de un comando.
- **rlogin:** permite el manejo remoto de máquinas de una red, automatizando *logins* entre máquinas que confían la una de la otra.
- **rexec:** permite la ejecución remota de comandos.
- **rwhod:** rwho es la versión de red de *who*, que ofrece información de los usuarios que están haciendo *log* en la actualidad. rwhod (el servidor rwho) sirve esta información a clientes rwho remotos.

Seguridad de servidor web

Apache es el servidor web, httpd, en la mayoría de distribuciones Linux.

httpd

Apache es el servidor HTTP más popular del mundo y ofrece muchos mecanismos de seguridad internos, incluyendo:

- Control de acceso de red basado en host (access.conf).
- Control sobre si los usuarios pueden y dónde pueden ejecutar *scripts* CGI (ExecCGI).
- Control sobre si los usuarios locales pueden y cuánto pueden sobre escribir sus configuraciones.

Ataques de denegación de servicio

Un ataque de denegación de servicio (DoS) es cualquier acción que incapacite el hardware, software, o ambos, de un host y que lleve a que no se pueda llegar al sistema y después deniegue el servicio de legitimar o deslegitimar usuarios. En un ataque DoS, el objetivo del atacante es sacar al host(s) de la Red.

Hay tres tipos de ataques DoS que ha sufrido Linux:

- Ataques DoS de hardware de red.
- Ataques en Linux trabajando en red.
- Ataques en aplicaciones Linux.

Cómo defenderse contra ataques DoS

- Desactivar la transmisión de direcciones.
- Filtre el tráfico ICMP, PING y UDP entrante.
- En servidores sacrificables y sin firewall, redefinir el tiempo muerto antes de que caiga una conexión abierta pero no resuelta. Esto reducirá los riesgos de tener ataques de cola de conexión, donde los atacantes inundan la cola de conexión al sistema con peticiones de conexiones abiertas.
- Activar la intercepción TCP del router. En la intercepción TCP es donde el router intercepta y valida las conexiones TCP. Las conexiones que no pueden llegar a un estado establecido después de un tiempo razonable, se cierran. También se cierran las que llegan de hosts inaccesibles. En ambos casos, el servidor solo engancha conexiones válidas y totalmente abiertas.
- Estar al día de los parches de los proveedores y de las actualizaciones de kernel.
- Utilizar filtros de paquetes para evitar direcciones de fuente sospechosas.

Firewalls

Un firewall es un dispositivo que filtra la información de entrada a una red privada. Normalmente, es un direccionador, una computadora autónoma con filtro de paquetes o software proxy, o un paquete de firewall (hardware que filtra y hace proxies). Un firewall puede

servir como punto de estrangulamiento (punto de entrada único a su sitio). El firewall evalúa las peticiones de conexión, y solo garantiza las de los hosts autorizados.

Los firewalls realizan algunas tareas, como:

- Filtro y análisis de paquetes: posibilidad de analizar paquetes entrantes de múltiples protocolos, realizando evaluaciones condicionales.
- Bloqueo de protocolo y contenido: permite bloquear contenido (Java, JavaScript, VBScript, Activex, etc.) e incluso formas de ataque particulares (programas con patrones de comando comunes a un ataque en particular).
- Autentificación y encriptación de usuario, conexión y sesión: utilización de algoritmos y sistemas de autentificación de usuarios.

Un firewall protege a una red al menos en dos de estos niveles:

- Quién puede entrar.
- Qué puede entrar.
- Dónde y cómo pueden entrar.

Existen dos tipos principales de firewall:

- Firewall a nivel de red o filtros de paquetes.
- Pasarelas de aplicaciones.

Firewall a nivel de red: filtros de paquetes

Los firewall a nivel de red son direccionadores con capacidad de filtrado de paquetes. Puede permitir o denegar acceso a su sitio basándose en varias variables como:

- Dirección de fuente.
- Protocolo.
- Número de puerto.
- Contenido.

Los firewall basados en direccionadores son soluciones de perímetro, es decir, son dispositivos

externos. Todo el tráfico exterior debe pasar a través del direccionador. Estos firewall pueden vencer al spoofing y a los ataques DoS, e incluso convertir a la red en invisible para el mundo exterior.

La siguiente figura muestra el esquema de un firewall basado en direccionador.

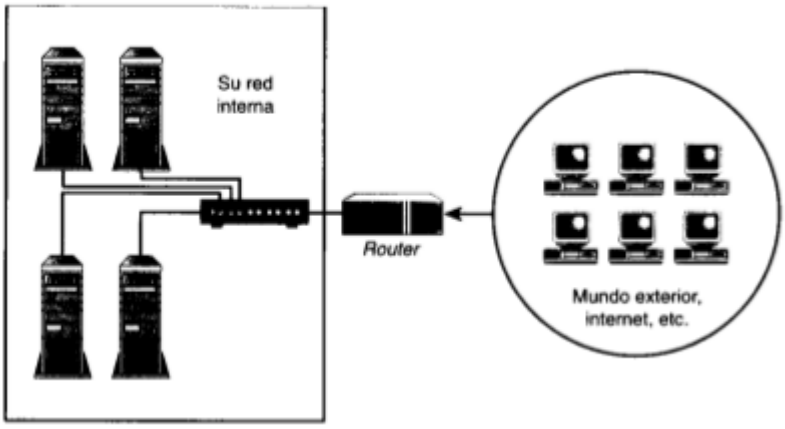

Figura 11.- Firewall basado en direccionador.

Algunos de los firewall basados en direccionadores son vulnerables a ataques y su actuación puede deteriorarse cuando utilice procedimientos de filtrado excesivamente estrictos.

Firewall de aplicación-proxy/pasarelas de aplicación

Las pasarelas de aplicación sustituyen a las conexiones entre los clientes externos y la red interna. Durante este cambio nunca se envían los paquetes IP. En su lugar, se produce una especie de traducción, actuando la pasarela de conducto y de intérprete. La ventaja es que se obtiene más control global sobre cada servicio individual y, en muchos casos, puede mantener la información del paquete. Una de las deficiencias de las pasarelas de aplicación es que se requiere configurar una aplicación proxy para cada servicio de la red.

Cómo evaluar si realmente necesita un firewall

Antes de instalar un firewall, debe pensar si en verdad lo necesita. Hay muchos entornos en los que un firewall no es adecuado, por ejemplo:

- Universidades. La investigación en las universidades a menudo la dirigen dos o más departamentos en colaboración (en segmentos de red separados) pueden ofrecer un acceso público limitado a sus alumnos. En tales entornos, es difícil trabajar bajo las fuertes restricciones de seguridad que conllevan los firewall.
- Proveedores de servicio de internet. Los clientes de ISP acceden a sus cuentas desde diferentes sitios. No se puede determinar de manera fiable cada dirección de IP desde la que puede provenir un cliente, no se puede mantener un control de acceso a nivel de firewall.

Los firewall son más adecuados para proteger redes privadas que necesitan acceso de salida a internet y ofrecen un acceso público de entrada mínimo y estrictamente controlado. Si esto no va con sus necesidades, aún puede disfrutar de un control de acceso a la red decente utilizando los TCP Wrappers.

tcpd: TCP Wrappers

Los TCP Wrappers son una de las herramientas más famosas del mundo para reforzar el control de acceso a la red. Se usa para filtrar el acceso de red a servicios de protocolos de internet. Permite que las direcciones IP, los nombres de terminales y/o respuestas de consultas de las terminales o subredes sean usadas como tokens sobre los cuales filtrar para propósitos de control de acceso. TCP Wrappers es lo más parecido a la funcionalidad del firewall que puede conseguir sin hacer uso de un filtro de paquete a escala total.

Logs y auditorías

Logging es cualquier procedimiento por el que un sistema operativo o aplicación graba eventos mientras ocurren y los guarda para un examen posterior. En un contexto de seguridad, el *logging* sirve para preservar un registro de las acciones dañinas de un atacante. Los *logs* ofrecen la única evidencia real de que ha ocurrido un ataque.

***Logging* en Linux**

El *logging* en Linux es dominante y sucede en los niveles de sistema, aplicación y protocolo. La mayoría de los servicios Linux imprimen información *log* en archivos estándar o en archivos de *log* compartidos. La mayoría reside en /var/log.

lastlog

lastlog sigue la pista de *logins* de usuario, da formato e imprime los contenidos del último *log login*, /var/log/lastlog. Se imprimirán el nombre de *login*, puerto y la última hora de *login*. El valor por defecto hace que las entradas *lastlog* se impriman en orden UID.

last

last informa del último *login* de usuarios. Busca a través del archivo /var/log/wtmp para mostrar una lista de todos los usuarios que hayan hecho *log in* (y *out*) desde que se creó el archivo.

Los datos suministrados incluyen:

- Usuarios.
- La terminal (o servicio) que utilizaron para hacer el *login*.
- Su dirección IP (o nombre del host) durante la sesión especificada.
- La fecha y hora.
- La duración de sus sesiones.

Cómo sortear los lastlog, last y wtmp

Los atacantes saben que /var/log/lastlog y /var/log/wtmp pueden descubrirlos. Por lo tanto, todo pirata mantiene un registro actualizado de barredores y limpiadores (programas que sortean y burlan los sistemas *logging* predefinidos). Algunos ejemplos de estos programas: cloak, cloak2, utclean, remove, utmpedit, SYSLOG Fogger, marry, etc.

La mayoría de los limpiadores de *log* requieren una de estas bibliotecas:

- utmp.h. Se utiliza para captar niveles de ejecución, cargar eventos de tiempo, procesos *init*, procesos *login*, procesos de usuario, tipo de *login*, nombre de host originario, etc.
- unistd.h. Se utiliza para captar mensajes de sistema sobre condiciones de error, condiciones de aviso, información de depuración, etc.

El atacante escribe código que abre utmp y, utilizando algo parecido a strncpy, reemplaza la línea actual con datos específicos de usuario, por un espacio en blanco o nada en absoluto. Para cubrirse contra piratas que manipulan sus entradas de *log*, debería utilizar una

herramienta de *logging* patentada o de terceros. Este método le ofrece dos ventajas, la primera, pocos piratas sabrán que está utilizando herramientas de *logging* especiales, segunda, éstas herramientas derivarán sus logs de manera independiente, sin utilizar logs de sistema operativo como índice inicial. Si más tarde se compara esta información con los *logs* de sistema por defecto y encuentra una discrepancia, se sabrá que se ha llevado a cabo una intrusión. Se puede también aislar los *logs* de la manipulación, escribiéndolos en medio de una sola escritura o en un servidor de *log* remoto.

Detección de intrusiones

La detección de intrusiones consiste en utilizar herramientas inteligentes y automáticas para detectar intentos de intrusión en tiempo real. Dichas herramientas se llaman Sistemas de Detección de Intrusiones (IDS).

Tipos de IDS

- Sistemas basados en normas. Basados en bibliotecas y bases de datos de ataques y firmas responsables de ataque conocidos. Cuando el tráfico entrante se encuentra con un criterio o norma particular, se etiqueta como un intento de intrusión. La desventaja de estos sistemas es que dependen del paso del tiempo (la base de datos de ataques debe ser actual). También, si una regla es demasiado específica, los ataques que son similares a ella pero no idénticos, pasarán.
- Sistemas adaptables. Estos emplean inteligencia artificial, no solo para reconocer firmas de ataque conocidas, sino para aprender nuevas. La principal desventaja es su elevado coste, son difíciles de mantener y requieren conocimientos avanzados de matemática y estadística.

Conceptos básicos de detección de intrusiones

En los IDS basados en normas hay dos métodos: prevención y reacción:

- En el método preventivo, funciona como un sniffer, cuando se detecta una actividad sospechosa (un flujo de paquetes particular), el sistema actúa de manera apropiada. Permite que su sistema responda mientras un atacante está planeando su asalto.
- En el método de reacción, la herramienta de detección de intrusiones observa sus logs. Le alerta del hecho de que acaba de suceder un ataque.

Se puede conseguir un modelo de reacción utilizando herramientas de seguridad estándar Linux:

- Utilice *LogSurfer* para buscar cierta actividad predefinida y amenazadora en los *logs*. Se puede definir la forma de respuesta de *LogServer* cuando encuentre esta actividad.
- Un *script* para añadir la dirección del atacante a *host.deny* para que tcpd niegue futuras conexiones.

El único inconveniente del *script* es la vulnerabilidad al *spoofing*. Las direcciones de origen no son fiables y fácilmente falsificables, habiendo la posibilidad de que el atacante utilice una dirección de origen distinta cada vez. El método preventivo también tiene algunos inconvenientes. Uno es la vulnerabilidad a los ataques de saturación, y otro, las limitaciones de hardware y software. En los ataques de saturación, el atacante supone que su IDS reaccionará de forma idéntica cuando se encuentre con un ataque idéntico, entonces, inunda su host con múltiples ejemplos del mismo ataque desde diferentes direcciones, incapacitando su IDS. En cuanto a las limitaciones de hardware y software, estas pueden obligarlo a elegir el análisis de tráfico en lugar del análisis del contenido. El análisis de tráfico procesa títulos de paquetes y no contenido, siendo insuficiente ante ataques de paquetes que contienen firmas de ataque.

Recuperación de desastres

La recuperación de desastres se centra en recuperarse después de que los datos se hayan destruido. La mayor parte del tiempo estos desastres ocurren por fuerzas mayores, errores inocentes, fallo mecánico o virus de software.

Pasos que hay que dar antes de construir una red Linux

Se tiene que planificar la posibilidad de un desastre antes de construir la red Linux. Los pasos que se deben seguir son:

- Normalización del Hardware. Para limitar los diferentes procedimientos de configuración.
- Normalización del Software. Un buen particionado y configuración de los servicios aumentarán las posibilidades de supervivencia.

- Copias de seguridad. Realizar copias de seguridad para la recuperación de desastres y por seguridad.

WEBMIN

Webmin es una herramienta de configuración de sistemas accesible vía web para sistemas Unix. Con él se pueden llegar a configurar aspectos internos de muchos sistemas operativos como usuarios, cuotas de espacio, servicios, archivos de configuración, apagado del equipo, etc., así como modificar y controlar muchas aplicaciones libres, como el servidor web Apache, PHP, MySQL, DNS, Samba, DHCP, entre otros. Webmin también permite controlar varias máquinas a través de una interfaz simple, o iniciar sesión en otros servidores webmin de la misma subred o red de área local.

Backtrack

Backtrack es una distribución GNU/Linux en formato LiveCD pensada y diseñada para la auditoría de seguridad y relacionada con la seguridad informática en general. Actualmente tiene una gran popularidad y aceptación en la comunidad que se mueve en torno a la seguridad informática. Se deriva de la unión de dos grandes distribuciones orientadas a la seguridad, el Auditor + WHAX. WHAX es la evolución del Whoppix (WhiteHat Knoppix), el cual pasó a basarse en la distribución Linux SLAX en lugar de Knoppix. La última versión de esta distribución cambió el sistema base, antes basado en Slax y ahora en Ubuntu. Incluye una larga lista de herramientas de seguridad listas para usar, entre las que destacan numerosos scanners de puertos y vulnerabilidades, archivos de exploits, sniffers, herramientas de análisis forense y herramientas para la auditoría Wireless.

Whoppix y WHAX

Whoppix es una distribución Live de linux que nació con la intención de proporcionar un entorno unificado para la auditoría de seguridad. Su nombre deriva de *White Hat Knoppix*.

WHAX está pensado para pruebas de seguridad y penetración de sistemas. Posee las últimas versiones de varias herramientas de seguridad. El cambio de nombre se debe a la migración del sistema base, originalmente Knoppix, ahora SLAX.

Herramientas

Backtrack le ofrece al usuario una extensa colección de herramientas completamente usables desde un Live CD o un Live USB por lo que no requiere una instalación para poder usarse. O bien, se ofrece la opción de instalar en un disco duro. Entre las herramientas ofrecidas se encuentran:

- Aircrack-ng, Herramientas para auditoría inalámbrica.
- Kismet, Sniffer inalámbrico.
- Ettercap, Interceptor/Sniffer/Registrador para LAN.
- Wireshark, Analizador de protocolos.
- Medusa, herramienta para Ataque de fuerza bruta.
- Nmap, rastreador de puertos.

Y una larga lista de otras herramientas, que se agrupan en 11 familias:

- Recopilación de Información.
- Mapeo de Puertos.
- Identificación de Vulnerabilidades.
- Análisis de aplicaciones Web.
- Análisis de redes de radio (WiFi, Bluetooth, RFID).
- Penetración (Exploits y Kit de herramientas de ingeniería social).
- Escalada de privilegios.
- Mantenimiento de Acceso.
- Forenses.
- Ingeniería inversa.
- Voz sobre IP.

En la figura se muestra una imagen de la distribución Backtrack 5 R1 (última versión).

Figura 12.- Imagen del sistema de auditoría Backtrack 5 R1.

CAPITULO 2

ESTATUS DEL SERVIDOR DE LA UDA

Se va a analizar el estatus del servidor de la Universidad del Azuay basándose en su seguridad y el tipo de servicios que proporciona:

El servidor maneja una velocidad de 25Mbps actualmente. Para el futuro se piensa incrementarla a 100Mbps.

Seguridad Física

En la actualidad, la universidad cuenta con un buen sistema de seguridad físico para lo que son sus servidores. El ingreso al cuarto donde están almacenados éstos es a través de biométricos. El cuarto tiene un piso antiestático y un sistema de ventilación mediante módulos posicionados entre los diferentes racks. Este sistema de ventilación recicla el aire caliente a la parte superior en donde se encuentran varias ranuras hacia el exterior para tratar de evacuar este aire. A parte de lo que es ventilación, existe un sistema de sensores para monitorear el funcionamiento de los equipos dentro de los racks. Sensores de movimiento, de luz para saber que está encendido y si existe algún tipo de alarma, temperatura, etc.

Antes de ver el resumen del estado del servidor de la universidad, hay que tener en cuenta la existencia de 2 servidores, uno para las notas de los alumnos y otro para lo que es administrativo y docente.

Contraseñas

Tanto para al servidor de notas como para el servidor de los docentes, no existe un acceso web, sino solo mediante intranet. El nivel de seguridad de las contraseñas es bajo, y no se aplica el sistema de *“expiración de contraseñas”*. El servidor tiene un programa para el control del nivel

de contraseñas, llamado *"John The Ripper"*, que obliga a los usuarios a crear cuentas con contraseñas de mínimo 8 dígitos.

MySQL

MySQL es un sistema de gestión de base de datos. En el servidor de la UDA se encuentra configurado de la siguiente manera:

- El manejo se realiza a través de internet mediante un programa llamado "*phpMyAdmin*".
- Solo existen claves para los usuarios administradores y no para todos los usuarios.
- Solo aplica encriptación a lo que son claves e información sumamente importante.
- El número de conexiones permitidas es mayor al que viene por defecto.

Apache

Apache es un servidor web HTTP. El estado de este servidor es el siguiente:

- La versión del servidor y otra información delicada están ocultas para los usuarios y no usuarios.
- El servidor está montado sobre la cuenta "*apache*" y el grupo de usuario "APACHE".
- No existe restricción de acceso al servidor.
- No utiliza "*mod_security*".
- El servidor mantiene permisos para los archivos y carpetas del mismo..
- Ejecución de CGI activada.
- Limitación de tamaño de petición por defecto.

MailScanner (incluyen ClammAV y SpamAssassin)

MailScanner es un sistema de seguridad para e-mail y un paquete anti-spam para servidores de MTA. Está diseñado para ejecutarse en los servidores de correo operados por las empresas y los ISP a fin de que todos sus usuarios y clientes se puedan proteger desde un mismo lugar.
Por otro lado, *SpamAssassin* es un programa para la detección de *spam*. Y por último, *ClammAV* es un software antivirus de código libre.

La configuración del *MailScanner*, *SpamAssassin* y *ClammAV* es la siguiente:

- El filtrado de direcciones se realiza mediante listas RBL.
- Se realizan pruebas a las cabeceras o a los cuerpos de los mensajes.
- El puntaje para saber si un correo es *spam* está definido en 3.
- Control de *phishing* a través de *Astaro.*

A parte de estos sistemas de seguridad, la universidad hizo la adquisición de un equipo de última tecnología llamado NetEnforcer, que integra protección completa de red, correo electrónico, navegación a través de una interfaz inteligente vía web.

VSFTP

VSFTP es un servidor FTP para sistemas *Unix*. Su estado es el siguiente:

- FTP anónimo se encuentra activo.
- Existe un control de encriptación y ancho de banda disponible a través de *Astaro*.

Dovecot 3

Dovecot es un servidor de IMAP y POP3 de código abierto para sistemas GNU/Linux, escrito pensando en seguridad. La configuración de este servidor es la siguiente:

- Permite conexiones *ssl.*
- Solo tiene claves para *ssl* en modo lectura.

SendMail

Sendmail es un popular “agente de transporte de correo” (MTA) en internet, cuya tarea consiste en encaminar los mensajes de correos de forma que éstos lleguen a su destino. Aquí se presenta su configuración:

- Utilizan *Sendmail* desde que se creó el servicio de correo.
- Las actualizaciones de este agente se realizan automáticamente a través del *Update Manager* de CENTOS.

En muchos sitios de internet se menciona que el servidor de correo, *Sendmail*, es muy complicado en cuanto a configuración y velocidad desempeñando su servicio. Aun así los administradores del servidor no piensan reemplazar este agente de correo. Recientemente hubo una importante mejora en *Sendmail* que tiene relación con este relativo defecto. Se creó la posibilidad de programar "macros" con la finalidad de sustituir algunos comandos bien descriptivos y transformarlos en acciones de mayor escala, para evitar entrar en detalles de programación y hacer más efectiva y fácil la configuración.

DNS

DNS es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función es traducir nombres inteligibles para los humanos en direcciones IP. El servidor DNS está configurado de la siguiente manera:

- La seguridad contra *DoS, Footprinting* (intercepción), *IPSpoofing* y redireccionamiento de información es manejada, en parte, por el *Allot Netenforcer.*
- No tienen un servicio DNS redundante.
- No mantienen doble configuración de cliente (dos servidores DNS, servidor e ISP).
- Tienen bloqueo de IP en la intranet.
- No existe encriptación de datos del servidor DNS.

SSH

SSH es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos

si tenemos un Servidor X corriendo. También permite copiar datos de forma segura, gestionar claves RSA para no escribir claves al conectar a los dispositivos y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH.

- Utilizan el puerto 22.
- Utilizan *PermitRootLogin* a través de IP.
- No tienen filtrado IP.
- No manejan usuarios, solo administración.

Firewall

El servidor utiliza como firewall el *iptables*, en específico, el *arno-iptables-firewall*. No se brindaron más detalles de la configuración de este tipo de *firewall iptables* por condiciones de seguridad del servidor de la universidad.

Scanners

La administración del servidor no utiliza scanners para detectar puntos débiles.

NetEnforcer de ALLOT

Los dispositivos de gestión de tráfico NetEnforcer® de Allot permiten vincular las políticas de la universidad con acciones específicas en la red para aumentar y controlar la productividad y la satisfacción de los usuarios. En redes empresariales, NetEnforcer optimiza la red WAN de manera que las aplicaciones clave de la empresa ofrezcan el rendimiento necesario para que ésta alcance sus objetivos.

Astaro Security Linux

Astaro Security Linux es una solución Linux integrada de protección contra toda clase de virus y piratería informática, avalado por Sun Microsystems que lo usa para la seguridad de sus redes,

formado por un sistema operativo seguro, con sofisticados filtros de paquetes, un anti-virus muy potente, unos filtros de contenidos, proxy diversos y uso de varios métodos de encriptación para acceso remoto VPN. Su administración sencilla pasa por el WebAdmin o, de forma remota, a través de la Web con encriptación SSL de 128 bits para más seguridad. El cortafuego ofrece la inspección Stateful dinámica de paquetes, detección de Portscans y protección anti-spoofing. Es extensible hasta 25 interfaces de red y soporta configuraciones de cortafuego y WAN de las más complejas.

El tráfico de datos está encriptado a través de un VPN (Virtual Private Network), de tal forma que hace la información inaccesible a terceros. En cuanto a los virus, Astaro brinda una protección eficaz con su anti-virus, enfrentándose a los mensajes no solicitados usando unos filtros y desafiando a los gusanos más cabezones. En cuanto a los proxy, filtran los contenidos y se combinan harmoniosamente con un filtrado de paquetes.

Como último comentario de estado del servidor, no se considera la filtración de “ping” hacia el servidor, considerado un ataque de DoS.

CAPÍTULO 3

DISEÑO DE RECOMENDACIONES DE SEGURIDAD PARA EL SERVIDOR DE LA UDA

En base al estudio del estatus del servidor de la Universidad del Azuay, se pueden formular algunas recomendaciones para mejorar la seguridad de éste.

1. Realizar la comprobación proactiva de contraseñas para mejorar la seguridad de las mismas y no tener ningún agujero de seguridad en este sentido.
2. Crear un programa shell que resetee las contraseñas de los usuarios cada cierto tiempo, con la finalidad de mejorar la seguridad ya que si se obtiene una contraseña para acceder el sistema, después de ese tiempo ya no sería nada útil.
3. Implementar un sistema contra incendios.
4. Programa para la revisión de los interfaces de red con *ifconfig* o *ifstatus*.
5. Utilizar un programa para comprobar la integridad de los archivos. Se recomienda *Tripwire*.
6. Se recomienda aplicar scanners de red para detectar puntos vulnerables del sistema.
7. Reemplazar el servidor de correo electrónico Sendmail, por otro llamado *Qmail*. Es más fácil de configurar y es mucho más rápido y sencillo.
8. Se recomienda usar *Backtrack 5* como sistema de auditoría para detectar puntos vulnerables de la seguridad del servidor y detectar posibles intrusiones.
9. Filtrar el tráfico de ping en el servidor (*DoS*).

CONCLUSIÓN

Analizando los diferentes servicios que presta el servidor de la Universidad del Azuay, se pudieron encontrar algunas vulnerabilidades para evitar la edición o robo de la información por parte de un atacante. Las recomendaciones fueron las siguientes:

1. Realizar la comprobación proactiva de contraseñas para mejorar la seguridad de las mismas y no tener ningún agujero de seguridad en este sentido.
2. Crear un programa shell que resetee las contraseñas de los usuarios cada cierto tiempo, con la finalidad de mejorar la seguridad ya que si se obtiene una contraseña para acceder el sistema, después de ese tiempo ya no sería nada útil.
3. Implementar un sistema contra incendios.
4. Programa para la revisión de los interfaces de red con *ifconfig* o *ifstatus*.
5. Utilizar un programa para comprobar la integridad de los archivos. Se recomienda *Tripwire*.
6. Se recomienda aplicar scanners de red para detectar puntos vulnerables del sistema.
7. Reemplazar el servidor de correo electrónico Sendmail, por otro llamado *Qmail*. Es más fácil de configurar y es mucho más rápido y sencillo.
8. Se recomienda usar *Backtrack 5* como sistema de auditoría para detectar puntos vulnerables de la seguridad del servidor y detectar posibles intrusiones.
9. Filtrar el tráfico de ping en el servidor (*DoS*).

Se espera que estas recomendaciones sean aplicadas al servidor de la universidad, y así asegurar la información y funcionamiento del mismo mejorando la confiabilidad de sus usuarios.

REFERENCIAS BIBLIOGRÁFICAS

- ANÓNIMO, Linux: Maximun Security, Ed. especial. NY, Prentice Hall, 2005.
- KRAGEN, Javier Sitaker, How to find Security Holes, 2002.
- VAN BIESBROUCK, Michael, CGI Security Tutorial, 1996.
- BELETSKY, Boris D., Debian Linux Installation & Getting Started, 1997.
- KOEHNTOPP, Kristan. The Linux Partition HOWTO, 1997.
- FIPS 74, Guidelines for Implementing and Using the NBS Data Encryption Standard, 1981.
- FIPS 46-2, Data Encryption Standard, 1993.
- BARAN, Kaye, BARAN, Suarez, Security Breaches: Five Recent Incidents at Columbia University, NY 1990.
- FELDMEIER, David C., KARN, Philip R., UNIX Password Security – Ten Years Later, Bellcore NJ, 1989.
- BELGERS, Walter, UNIX Password Security, Diciembre, 1993.
- BISHOP, Matt, KLEIN, Daniel, Improving System Security via Proactive Password Checking, Computers and Security, 1995.
- BISHOP, Matt, A Proactive Password Checker, in Information Security: Proceedings of the IFIP TC11 Seventh International Conference on Information Security: Creating Confidence in Information Processing, D. T. Lindsay and W. L. Price (eds.), North-Holland, New York, NY (1991).
- RITTER, Terry, 2x Isolated Double-DES: Another Weak Two-Level DES Structure, Ritter Software Engineering, 1994.
- KIM, Gene H., SPAFFORD, Eugene H., The Design and Implementation of Tripwire: A File System Integrity Checker, 1994.
- TURNER & CHEN, MD5 and HMAC-MD5 Security Considerations, RFC 6151, 2011.
- MORRIS, Robert, A Weakness in the 4.2BSD UNIX TCP/IP Software, Bell Labs, 1985.
- FARROW, RIK, Sequence Number Attacks, 2001.
- COSTALES, Bryan, ALLMAN, Eric, RICKERT, Neil, The Sendmail Nutshell Book, 1993.
- COBB, Chey, COBB, Stephen, Denial of Service, CISSP, 1998.
- HARE, Chris, SIYAN, Karanjit, Internet Firewalls and Network Security (Second Edition), Prentice Hall, New Riders, 1996.

REFERENCIAS ELECTRÓNICAS

- Internet World, Sniffers and Spoofers, Disponible: http://www.internetworld.com/print/monthly/1995/12/webwatch.html, 1995.
- Domain Name Service Vulnerabilities, Consultado: 19/09/2011, Disponible: http:// *ciac*.llnl.gov/*ciac* /bulletins/ g-14.shtml, CIAC, 2001.
- Distribución para la auditoría de seguridad, Consultado: 21/10/2011, Disponible: http://www.backtrack-linux.org/.
- Herramienta de configuración de sistemas accesible vía web para sistemas Unix, Consultado: 21/10/2011, Disponible: http://www.webmin.com/.

ANEXOS