Seguridad en Sistemas Operativos Linux

Alexander Urrego Morera, Universidad Piloto de Colombia

***Resumen*—Asegurar cualquier servidor sin importar el sistema operativo con el que funcione es de suma importancia, debido al aumento de ataques cibernéticos que se presentan hoy en día con el fin de robar información, destruir información u obtener acceso a datos que después pueden ser vendidos y adquirir algún tipo de beneficio económico o de otra índole. Este documento es una guía sobre cómo proteger un computador o servidor con sistema operativo Linux, fortalecer y asegurar cualquier distribución Linux, pretende acercar al lector un poco más al funcionamiento y estructura del Kernel Linux. Se debe conocer que se quiere asegurar y por qué debemos asegurarlo. Mostrar lo Seguro, confiable que puede ser un servidor con cualquier distribución Linux instalada donde se apliquen correctamente las técnicas para su aseguramiento y mantenimiento que se ejecutan por el administrador del servidor diariamente para garantizar que la información y aplicaciones estén salvaguardadas.**

***Abstract*—Securing any server regardless of the operating system with which it works is of utmost importance, due to the increase in cyber-attacks that occur today to steal information, destroy information or gain access to data that can later be sold and acquired. some type of economic or other benefit. This document is a guide on how to protect a computer or server with a Linux operating system, strengthen and secure any Linux distribution, it aims to bring the reader a little closer to the operation and structure of the Linux Kernel. You must know what you want to insure and why we should insure it. Show how safe, reliable a server can be with any Linux distribution installed where the techniques for its assurance and maintenance are correctly applied, which are executed by the server administrator daily to guarantee that the information and applications are safeguarded**

***Index Terms*—Kernel, Hardering, VPN, Backup**

I. INTRODUCCIÓN

LOS servidores y equipos de escritorio instalados con Sistemas Operativos Linux cada día van en aumento, las distribuciones Linux se encuentran en un estado de madurez y robustez importante que permite que se puedan utilizar para cualquier ambiente de trabajo. Los entornos se han vuelto más amigables para el usuario, su facilidad de instalación, su practicidad al momento de usar el sistema operativo y la gran cantidad de paquetes que el usuario puede instalar para administrarlo a hecho que el porcentaje en el mercado aumente cada día más; y la brecha con los equipos que cuentan con sistema operativo Windows sea cada vez menor. Existen comunidades que se dedican exclusivamente a desarrollar y dar soporte en aplicaciones para sistemas operativos Linux. Sin nombrar la gran cantidad de distribuciones que existen de acuerdo con la necesidad de los usuarios y las funciones que esta distribución realizara; Según estadísticas del año 2020 el 70% de los servidores a nivel mundial utiliza sistema operativo Linux para su funcionamiento. Para mantener el sistema operativo Linux lo más actualizado posible como un sistema operativo Windows Server donde Microsoft empresa desarrolladora libera actualizaciones cada vez que se encuentra una vulnerabilidad crítica, es una tarea que se debe realizar diariamente por el administrador de servidores o el usuario final. Se debe estar revisando foros de seguridad o estar suscrito a los boletines de cada distribución que se tenga instalada, donde se informan las vulnerabilidades encontradas y la forma de remediarlas. Es una medida poco efectiva pero necesaria, ya que mientras se actualiza el boletín de seguridad, se recibe y analiza la vulnerabilidad el impacto que pueda, posteriormente se aplica el parche de seguridad o se corrige la vulnerabilidad puede pasar demasiado tiempo y mientras tanto nuestro servidor o equipo puede ser vulnerado por un atacante. Adicionalmente de las vulnerabilidades que se debe cubrir del Sistema Operativo, se debe también mantener actualizados los paquetes a aplicativos que se tienen instalados en el servidor. Lo ideal sería parchear cada nueva vulnerabilidad inmediatamente después de que sea descubierta. En la vida real esto es casi imposible; un administrador de sistemas responsable de la gestión de parcheo de los sistemas operativos Linux se enfrentan a múltiples inconvenientes para poder aplicar parches, incompatibilidad con aplicaciones ya instaladas, falta de documentación. En el caso de los sistemas operativos Linux por ser un sistema operativo libre la liberación de parches puede demorar mientras la comunidad documenta y corrige el error en la distribución correspondiente.

Casi el 99% de los sistemas de cómputo se encuentran conectados a Internet, y en el momento que un equipo esté conectado a Internet ya lo convierte en un equipo vulnerable. En el caso de las distribuciones Linux están entre los productos con más vulnerabilidades encontradas en los últimos 10 años.

Para lograr que cualquier Sistema Operativo sea seguro uno de los primeros pasos que se debe realizar es conocer el sistema operativo, sus componentes, su funcionamiento, no se puede asegurar algo de lo cual no se conoce, no se sabe administrar.

II. DISTRIBUCIONES LINUX

Una distribución de Linux es una versión personalizada del sistema operativo, en la actualidad existen demasiadas distribuciones Linux las cuales serán imposibles de enumerar algunas ya desaparecidas o absorbidas por otras distribuciones u otras empresas. Otras solo conocidas en una comunidad muy pequeña y por supuesto las grandes y populares distribuciones que abarcan el mercado. Hay un equipo de trabajo que regula el código que se integra en cada versión del Kernel de Linux.

Documento Creado para optar al grado de Especialista en Seguridad Informática de la Universidad Piloto de Colombia. Alexander Urrego Morera, Ingeniero de Sistemas de la Escuela Colombiana de Carreras Industriales 'ECCI', Estudiante de la especialización en Seguridad Informática en la Universidad Piloto de Colombia (alexander-urrego@upc.edu.co)

Fig. 1: Productos con más vulnerabilidades Técnicas.

La condición de código abierto que trabaja bajo la licencia GNU/Linux permite que cualquier persona pueda modificar el Kernel de Linux y generar su propia versión con un propósito especifico o ajustarlo de acuerdo con una necesidad especifica. Existen versiones de escritorio donde facilitan las tareas que realiza el usuario, herramientas ofimáticas y de diseño, navegación en Internet. Y las versiones para Servidores las cuales están diseñadas para funcionar 7x24, por lo general se instalan sin ambiente gráfico para que sean más eficientes, mejorar su rendimiento y ser administrados remotamente. También encontramos muy pocas versiones de Linux empresariales las cuales son pagas y una de sus grandes ventajas es que tienen soporte Técnico sobre la distribución como lo son Red Hat.

III. NIVELES DE SEGURIDAD

A. Niveles de Seguridad del TCSEC

Para poder determinar si un sistema operativo Linux está completamente asegurado, se han definido varios estándares a nivel mundial para certificar la seguridad de los productos informáticos. El estándar TCSEC (Trusted Computer System Evaluation Criteria) orange Book 2, es uno de los estándares más utilizados a nivel mundial; Fue desarrollado en 1983 por el Departamento de Defensa de los Estados Unidos.

Este sistema contempla varios niveles de seguridad, cada uno de los cuales se cumple el nivel anterior. Los criterios para evaluar los niveles de seguridad tienen en cuenta la factibilidad de que se presente un ataque al hardware, software o la información almacenada en estos sistemas. También se toman en cuenta los niveles de seguridad física, autenticación de usuario, confiabilidad del software, Sistema Operativo y las aplicaciones:

- Nivel D1:El sistema en su totalidad no es confiable y no cumple con ninguna especificación de seguridad. No existe protección de hardware ni autenticación de usuarios. Estos sistemas no tienen control de acceso sobre los archivos. Y no se define quien realiza la acción sobre él.
- Nivel C1: El nivel C cuenta con 2 subniveles de seguridad: C1 y C2, En el nivel C1 se implementa un acceso de control discrecional y la identificación y autenticación de usuarios. Se implementa así la distinción entre usuarios genéricos y administradores del Sistema. Se puede determinar control de acceso a los recursos del Sistema, archivos y directorios. En el caso de los sistemas operativos Linux y Unix algunas tareas del sistema únicamente pueden ser ejecutadas por el usuario root, en el caso de los servidores centralizados se puede dar el caso de que más de una persona utilice el usuario root para ejecutar tareas de administración.
- Nivel C2: El nivel C2 fue diseñado para ayudar a solucionar los inconvenientes que se presentan en el subnivel C1, Este subnivel incluye una característica adicional que crea un medio de acceso controlado. Este medio tiene la capacidad de reforzar las restricciones a los usuarios en la ejecución de algunos comandos o acceso a algunos archivos. Este nivel de seguridad requiere de auditoría del sistema. La auditoría se utiliza para mantener registro de todos los eventos relacionados con la seguridad, como son las actividades realizadas por el administrador del Sistema.
- Nivel B1: El nivel B cuenta con 3 subniveles. El nivel B1, o protección de seguridad encriptada, es el primer nivel que soporta seguridad de multinivel, Este nivel tiene como principio de que un objeto bajo control de acceso obligatorio no puede aceptar cambios en los permisos realizados por el dueño del archivo.
- Nivel B2: Conocido como protección estructurada, requiere que se etiquete cada objeto. Los dispositivos como discos duros, cintas o terminales podrán tener asignado un nivel sencillo o múltiple de seguridad. Este es el primer nivel que se refiere al problema de un objeto de un nivel más elevado de seguridad en comunicación con un objeto de un nivel inferior.
- Nivel B3: Nivel B3 o nivel de dominios de seguridad, refuerza a los dominios con la instalación de hardware. Este nivel requiere que la terminal del usuario se conecte al sistema por medio de una ruta de acceso segura.
- Nivel A: Implementa un proceso de diseño, control y verificación mediante métodos matemáticos. Es el nivel de seguridad más elevado. Para lograr este nivel de seguridad, todos los componentes de los niveles inferiores deben incluirse, el diseño requiere ser verificado en forma matemática; [1].

Los niveles de seguridad del TCSEC hoy en día son obsoletos. Sin embargo, han representado una importante base sobre la cual se han desarrollado posteriormente los principales estándares como lo son: el europeo ITSEC (Information Technology Security Evaluation Criteria) en 1991, el canadiense se CTPEC (Canadian Trusted Computer Product Evaluation Criteria) en 1993 y el estadounidense FCITS (Federal Criteria for Information Technology Security), también en 1993.

B. Niveles de seguridad del Common Criteria

El estándar o norma que ha conseguido unificar la seguridad de los productos informáticos a nivel internacional es la ISO/IEC 15408, más conocida como Common Criteria. Esta norma inicio en 1990 por la Organización Internacional de Estandarización – ISO y es reconocida por los 27 países firmantes del Common Criteria Recognition Agreement. El objetivo de esta norma es evaluar las funciones de seguridad y el nivel de confianza de un producto IT.

Un concepto clave de este estándar son los perfiles de protección, que definen un conjunto de requerimientos y objetivos de seguridad para una categoría de productos, con similares necesidades por parte del usuario. Además, se definen objetivos de seguridad para los distintos productos y sistemas informáticos que van a ser evaluados. Además, Common Criteria define una escala de siete niveles de confianza (EAL – Evaluation Assurance Levels) para medir los perfiles de protección y los objetivos de seguridad.

- EAL1: funcionalidad probada
- EAL2: estructuralmente probado
- EAL3: probado y verificado metódicamente
- EAL4: diseñado, probado y revisado metódicamente.
- EAL5: diseñado y probado semi formalmente
- EAL6: diseñado, verificado y probado semi formalmente
- EAL7: diseñado, verificado y probado formalmente

Estos niveles se han diseñado para ser compatibles con los estándares previos, como el TCSEC e ITSEC.

C. Niveles de Seguridad del TCSEC

ITSEC o "White Book" es un esfuerzo conjunto de la Unión Europea (UE) para generar un criterio de evaluación de seguridad estandarizado para la región. La ITSEC evalúa productos y sistemas; Concede a los productos que se evalúan satisfactoriamente niveles de seguridad de E1 hasta el E6, siendo el nivel E1 el nivel más bajo.

El ITSEC se aplica a un TOE (target of evaluation), que es el sistema o producto que se someterá a un análisis y evaluación en cuanto a su seguridad. El TOE contiene dos elementos:

1) Un conjunto de componentes del sistema que contribuyen directamente a satisfacer los objetivos de seguridad.
2) Un conjunto de componentes que no se encargan de cumplir la seguridad, pero que deben estar presentes para su funcionalidad de forma correcta para que su seguridad se cumpla.

El sistema evaluado, llamado blanco de la evaluación, es sujeto a una examen minucioso de sus características de seguridad que culminan en la prueba funcional y de la penetración comprensiva e informada. Los niveles de seguridad de ITSEC son:

- Nivel E6: Un plan detallado de procedimientos para recuperación.
- Nivel E5: Sistema muy seguro, la auditoría es otra exigencia. Política de seguridad estricta.
- Nivel E4: Plan de seguridad. El control de acceso es obligatorio y un responsable de seguridad debe decidir qué cierta información solo puede ser accedidas por personas de confianza.
- Nivel E3: Sistemas en los que se exige palabras de acceso y se verifica con auditorías.
- Nivel E2: Medidas discrecionales para proteger los datos de los intrusos.
- Nivel E1: Sistema que no ha implementado ningún nivel de seguridad.

IV. EL KERNEL

EL Kernel de Linux o núcleo del sistema operativo es el componente más importante del sistema operativo se encuentra cargado en la RAM al iniciar el sistema. Trabaja en modo privilegiado del procesador, no tiene ninguna restricción a la hora de ejecutar cualquier proceso. Utiliza todas las instrucciones del procesador, direcciona la memoria, accede directamente al hardware, al modificar el Kernel erróneamente o realizar cualquier actividad puede ser fatal para el funcionamiento del sistema operativo. Que es lo que conocemos como "Kernel Panic" equivalente a la pantalla azul en los sistemas operativos Windows.

Su estructura puede separarse en una serie de componentes, o módulos de gestión orientados a:

- Gestión de procesos: qué tareas se van a ejecutar y en qué orden y prioridad. Un aspecto importante es la planificación de la CPU.
- Intercomunicación de Procesos y sincronización.
- Gestión de entrada/salida (E/S): control de periféricos y gestión de recursos asociados.
- Gestión de memoria: optimización del uso de la memoria, sistema de paginación y memoria virtual.
- Gestión de ficheros: cómo el sistema controla y organiza los ficheros presentes en el sistema, y accede a los mismos. [2]

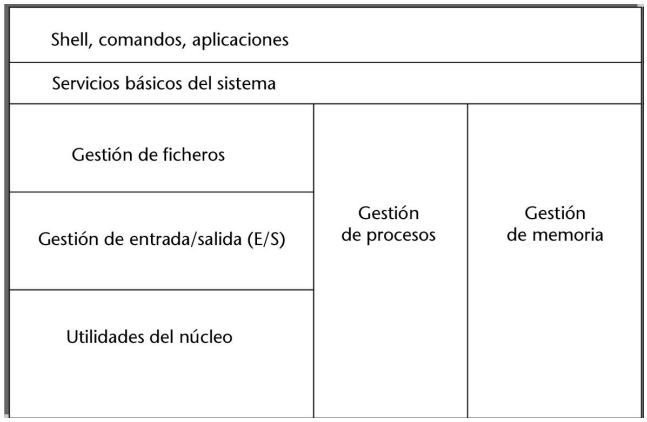

Fig. 2: Funciones básicas del Kernel , respecto a las aplicaciones y comandos ejecutados.

Al ser de libre acceso el código fuente de los sistemas operativos Linux, cualquier persona puede dedicarse a leer las líneas de código del Kernel, con el objetivo de encontrar fallos

y explotarlos. Pero es una tarea demasiado larga y tediosa; Y mientras alguien está leyendo el código fuente con el fin de vulnerarlo existen personas que están leyendo el código fuente con el fin de repararlo y generar parches de seguridad con el fin de minimizar esa brecha de seguridad.

El Kernel de Linux está diseñado para ser multiusuario, lo que nos indican que varios usuarios pueden ejecutar programas en paralelo. Y cualquier usuario puede ejecutar cualquier comando sin importar lo que estén realizando los demás usuarios. El kernel también tiene como función proporcionar mecanismos de control de acceso para evitar que varios usuarios o procesos pueden ejecutar un mismo proceso varias veces.

V. HARDERING EN SISTEMAS OPERATIVOS LINUX

El hardering consiste en el endurecimiento del sistema operativo, con el fin de reducir y evitar amenazas a los que está expuesto. El proceso de reducción de vulnerabilidades se consigue estableciendo medidas y políticas de seguridad con el objetivo de estar preparados a un ataque informático de cualquier tipo u daño en el mismo sistema operativo, garantizando la integridad, disponibilidad y confidencialidad de la información. A continuación, se mencionarán un conjunto de buenas prácticas, con el fin de que nuestro sistema operativo Linux sea más seguro:

A. Conexión remota SSH

SSH o Secure Shell, es un protocolo de administración remota que permite a los usuarios controlar y modificar un servidor de forma remota a través de Internet o de una red local. Proporciona un mecanismo para autenticar un usuario remoto, el servicio se creó como reemplazo seguro para el telnet sin cifrar y utilizar técnicas criptográficas con el fin de garantizar que toda la comunicación desde y hacia el servidor se realizan de manera encriptada.

Por defecto el servicio SSH utiliza el puerto 22 para establecer la conexión. Es recomendable cambiar este número de puerto para evitar que ciberdelincuentes puedan intentar iniciar sesión, o pasar desapercibidos a un escaneo de puertos que se pueden realizar desde internet o desde cualquier red LAN. En el fichero de configuración debemos modificar el parámetro port y asignarle un puerto que deseemos que escuche diferente al puerto 22, por ejemplo: “Port 64280”.

Se debe bloquear el acceso al usuario root en las conexiones remotas, siempre que alguien intenta atacar un servidor por defecto utiliza el puerto 22 y el usuario root, Desactivando el usuario root y utilizando el comando “sudo” para elevar privilegios a superusuario nos permitirá dotar de mayor seguridad al servidor. El parámetro que se debe modificar es: “PermitRootLogin no”. De esta manera las conexiones remotas con el usuario root quedaran bloqueadas por defecto evitando ataques de fuerza bruta contra el servidor SSH intentando adivinar el password del usuario root.

Otros parámetros de configuración que se pueden recomendar aplicar a la configuración del servidor SSH son:

- LoginGraceTime: Tiempo de espera para introducir la contraseña evitando que el atacante pueda pensar mucho la contraseña a introducir.
- MaxAuthTries: Numero de intentos permitidos al introducir la contraseña errónea antes de desconectar la sesión.
- AllowUsers: Es crear una lista blanca de usuarios permitidos para acceder al servidor. Una medida muy restrictiva, pero a la vez muy segura ya que bloqueara todas las conexiones de los usuarios que no estén en el listado.
- MaxStartups: Numero de logins simultáneos desde una IP, para evitar que se presente un ataque de fuerza bruta con varias sesiones a la vez desde una misma ip.
- DenyUsers: Se crea una lista negra de usuarios que no les está permitido la conexión remota.
- AllowGroups/DenyUsers: Exactamente igual a la anterior, pero en lugar de crear una lista blanca/negra de usuarios, es de grupos de usuarios.

Otra buena practica es restringir el rango de direcciones IP que puedan acceder al servidor por SSH. Permitiendo únicamente conexiones de redes de confianza, como a la red local o direcciones IP Publicas conocidas.

Una forma de restringir desde que IP’s se puede acceder el servidor es configurando TCP Wrappers. En el archivo hosts.allow se indica las redes con acceso. Se debe editar el siguiente archivo:
/etc/hosts.allow donde se indica desde cuál o cuáles redes se tiene acceso:
sshd : 10.72.18.0/24,172.16.48.0/24
Se Utiliza la coma (,) para separar varias redes.
La conexión con clave publica para realizar una conexión ssh es un método aun más seguro , En el cual los usuarios no necesitan recordar la contraseña. Funciona con dos claves una publica y otra privada. El servidor permitirá la conexión cuando la llave coincida con la del servidor.

Para realizar esta configuración en los sistemas operativos Linux se deben generar las claves en el equipo cliente de la siguiente manera:
ssh-keygen, con la opción “-b” indica el numero de bits de la llave, y la opción -t el tipo de cifrado que deseamos (rsa, dsa,)

```
alexander@alexander:~$ ssh-keygen -b 2048 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alexander/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alexander/.ssh/id_rsa
Your public key has been saved in /home/alexander/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:waugOw7+h3e24NQ/0iHgRiZ80AzXBY6ZSZ0S1k3i3AE alexander@alexander
The key's randomart image is:
+---[RSA 2048]----+
|    .+==EB+      |
|    .++O*o..     |
|   . .=.+o.      |
|    o =   o      |
|     B . S       |
|    . +.o .      |
|. . oo..o .      |
|...ooo.+.o       |
| o+oo.o.o..      |
+----[SHA256]-----+
```

Fig. 3: Creación de llaves.

Después de generar las llaves , se generan 2 archivos:

- id_rsa es la clave privada, la que permanecerá en la máquina local.
- id_rsa.pub es la clave pública, la que se debe copiar al servidor remoto al que se quiere acceder.

En el servidor al cual se va acceder remotamente se debe copiar el contenido del archivo id_rsa.pub dentro del archivo "authorized_keys". Este fichero en muchas ocasiones no se encuentra, se puede crear manualmente. Si deseamos que todos los equipos se conecten al servidor por este medio y no por medio de una contraseña se debe modificar la configuración en el archivo "/etc/ssh/sshd_config" y en el parámetro "PasswordAuthentication" y cambiamos su valor a no, lo cual el servidor no aceptara ninguna contraseña para recibir conexiones. Se pueden agregar tantas llaves publicas como deseemos dependiendo de la cantidad de usuarios que accedan al servidor. Los usuarios que se conecten remotamente a nuestro servidor también se podrán restringir las tareas y comandos que podrán ejecutar en el servidor.

B. Listar puertos abiertos

Se debe realizar un listado de todos los puertos que se encuentran abiertos en el servidor y depurar que puertos se necesitan dejar abiertos y que puertos no necesitamos que estén abiertos. En los sistemas operativos Linux cada puerto se habilita dependiendo del proceso que está ejecutando y el acceso al puerto depende del servicio que preste.

Para listar los puertos que se encuentran escuchando o se encuentran abiertos en el servidor para recibir peticiones se debe ejecutar el comando "netstat -tupln".

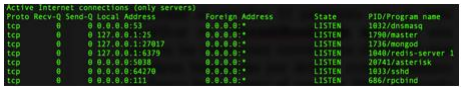

Fig. 4: Ejemplo de puertos abiertos.

Después que se determine que puertos son necesarios dejar abiertos permanentemente en el servidor, se deben realizar algunas prácticas como se vio en el punto anterior de cambiar el puerto predeterminado por un puerto poco conocido, Ejemplo cambiar el puerto 22 para la conexión SSH por el puerto 64280. Y restringir el acceso al puerto únicamente las direcciones IP necesarias, por ejemplo en un servidor de base de datos, las únicas direcciones IP que deben acceder a este puerto es el de los servidores de aplicaciones.

C. Habilitar IPTABLES

debe configurar un firewall o cortafuegos en el servidor, para restringir el acceso a los puertos y servicios que deben quedar abiertos únicamente a las direcciones IP que deben consumir el servicio.

Iptables es un firewall incluido en el kernel de los servidores con sistemas operativos Linux. Su funcionamiento se basa en aplicar reglas que el mismo firewall ejecuta. Filtra o restringe el tráfico de red que pasa por el servidor.
Existen 2 maneras de implementar un IPTABLES:

- Política por defecto Aceptar todo: Por defecto Acepta todo el tráfico que entra y sale del servidor, solo se denegara lo que se especifique en las reglas que se cree.
- Política por defecto Denegar: Todo es denegado, y solo se permitirá pasar por el firewall lo que se acepte explícitamente en las reglas que se cree . Esta Política es la más utilizada, para bloquear todo excepto [Reglas].

Se puede especificar la dirección IP de origen, dirección IP de destino, protocolo TCP/UDP. Iptables gestiona, mantiene e inspecciona las reglas de filtrado de paquetes IPv4 a través de tablas. Estas tablas clasifican y organizan las reglas de acuerdo con el tipo de decisiones que se deben tomar sobre los paquetes. Cada tabla contiene un numero de cadenas (chains), algunas predeterminadas y otras escritas por el administrador del servidor o del firewall, las cadenas representan los eventos que disparan o inician a cada regla (hooks de netfilter). De esta manera, las cadenas determinan cuando las reglas son evaluadas (cuando un paquete ingresa al sistema, cuando un paquete sale del sistema, cuando un paquete se debe reenviar hacia otro sistema, etc.).

Los diferentes hooks presentes en el framework netfilter coinciden con las siguientes cadenas de iptables:

- PREROUTING: Trafico entrante, las reglas de esta cadena son procesadas antes de tomar cualquier decisión de ruteo respecto hacia donde enviar el paquete
- INPUT: Trafico entrante luego de haber sido ruteado y destinado al sistema local.
- FORWARD: Trafico entrante luego de haber sido ruteado y destinado hacia otro host (reenviado).
- OUTPUT: Trafico saliente originado en el servidor.
- POSTROUTING: Trafico saliente originado en el servidor o reenviado, luego de haber sido ruteado.

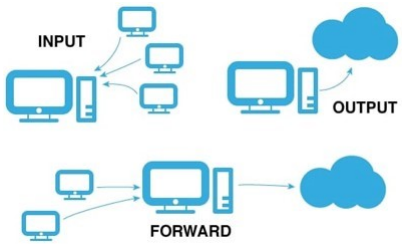

Fig. 5: Funcionamiento firewall.

D. Ataques DDOS

Es imposible listar los ataques que se pueden realizar sobre una red, ya que pueden variar de un momento a otro. El ataque más común que se realiza sobre los servidores que se encuentran expuestos a Internet son los denominados ataques de denegación de Servicios (Dos).

En un ataque de denegación de servicios, se intenta evitar que los usuarios accedan a un servidor o servicio. Estos son generados en la mayoría de sus casos a través de bots (Equipos zombis que se conectan simultáneamente a un servidor). Su función es saturar el ancho de banda de información inservible con el fin de que los servidores no puedan recibir ni procesar mas información.

La forma de contrarrestar los tipos de ataques Dos más comunes es la siguiente:

- ICMP Attack: Este tipo de ataque permite agotar el ancho de banda de la víctima. Consiste en enviar una gran cantidad de información enviando paquetes ICMP Echo Request.

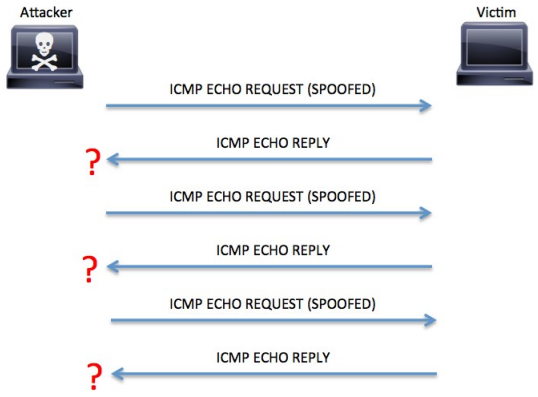

Fig. 6: Ataque ICMP Echo Request.

Para mitigar este tipo de ataques es necesario deshabilitar el ping en nuestro servidor, hacer nuestro servidor invisible ante escaneos. Se debe agregar la siguiente linea en el archivo /etc/sysctl.conf. net.ipv4.icp_echo_ignore_all= 1

- SYN Flood: Consiste en enviar paquetes TCP con el Flag SYN activado, con el objetivo de enviar cientos de paquetes a un servidor y abrir diferentes conexiones , con el objetivo de saturar el servidor por completo, este tipo de ataques se realiza utilizando una dirección IP falsa.

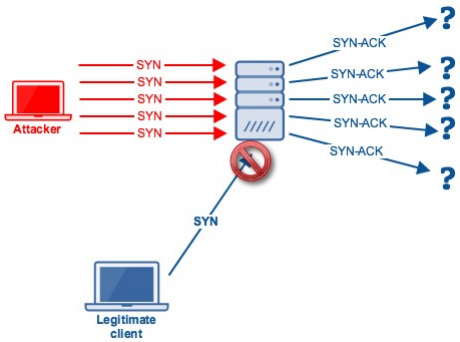

Fig. 7: Ataque Syn Flood.

Para mitigar este tipo de ataques se debe modificar los siguientes parámetros en el archivo /etc/sysctl.conf.

- Habilitar las cookies syn, logrando que el servidor se comporte como si se ampliara la cola de conexiones syn.
  net.ipv4.tcp_syncookies = 1
- Aumentar la cola de trapajos pendientes de la cola SYN. El tamaño predeterminado es 1024. net.ipv4.tcp_max_syn_backlog = 2048
- Reducción de reintentos syn_ack, Ajustar el parámetro de Kernel tcp_synack_retries para que el Kernel cierre las conexiones de estado SYN_RECV, el valor predeterminado es 5. net.ipv4.tcp_synack_retries = 3
- Configurar el tiempo de espera de SYN_RECV , ayudara a reducir el ataque de inundación SYN. El valorpredeterminado es 60, lo podemos reducir a 30. net.ipv4.netfilter.ip_contrack_tcp_timeout_syn_recv = 30
- Prevención de falsificación de la dirección IP. Nos ayudara a proteger el servidor de suplantación de direcciones IP.
  net.ipv4.conf.all.rp_filter = 1

- UDP Flood: Se utiliza el protocolo de conexión UDP para enviar una gran cantidad de paquetes al servidor utilizando muchas conexiones al mismo tiempo, ocasionando que los recursos (Memoria RAM, Procesador) del servidor sean insuficientes para manipular y procesar tal cantidad de información, en consecuencia, el sistema se bloquea.

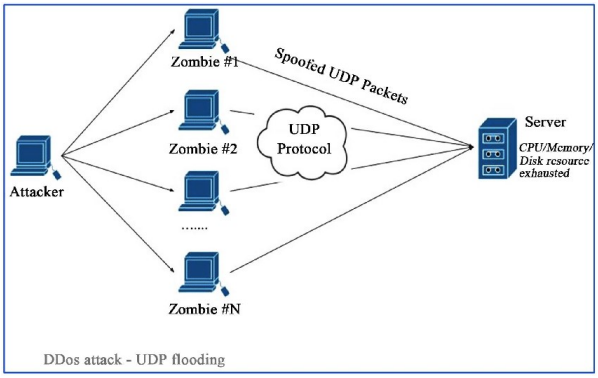

Fig. 8: Ataque UDP Flood .

Una de las maneras de mitigar los ataques UDP Flood es por medio del iptables con las siguientes reglas:

- Crear la cadena para UDP flood
  iptables -N cadena-udp-flood
- Saltar a la cadena cuando el UDP es detectado
  iptables -A INPUT -p udp -j cadena-udp-flood
- Limitar la velocidad UDP a 10/segundos con limite de 20 iptables -A cadena-udp-flood -m limit – -limit 10/s – -limit-burst 20 -m comment – -comment “Limite velocidad UDP” -j RETURN
- Logear
  iptables -A cadena-udp-flood -m limit – -limit 6/h – -limit-burst 1 -j LOG – -log-prefix “Probable udp flood “
- Baneados durante 5 minutos los que más floodean
  iptables -A cadena-udp-flood -m recent – -name blacklist_300 – -set -m comment – -comment “BlackList origen IP” -j DROP

E. Software Instalado

Se debe listar todos los paquetes instalados en el servidor para mantener instalados únicamente los paquetes que son necesarios para su funcionamiento o prestar el servicio para el cual ha sido configurado, por ejemplo un servidor exclusivo de base de datos, debería tener instalado el paquete del motor de base de datos el cual se está utilizando y no se debería instalar ningún otro paquete, servidor Web, servidor de archivos el cual no es la funcionalidad principal para lo que ha sido configurado el servidor.

Deshabilitando los paquetes que no son necesarios para el funcionamiento se está reduciendo la superficie de ataque en un servidor.

No se deben instalar paquetes de fuentes desconocidas o poco confiables, se debería acceder al código fuente desde la web oficial del paquete que se desea instalar. La mayoría de paquetes que se necesitan instalar para el funcionamiento de un servidor vienen incluidos en los puertos por defecto para ser instalados.

F. Actualización Kernel y Parches de Seguridad

Después de la instalación de cualquier Sistema Operativo, la primera actividad que se debe ejecutar es actualizar todos los repositorios para que al momento de instalar cualquier paquete se esté asegurando que es la última versión liberada. Las actualizaciones se deben realizar utilizando el manejador de paquetes nativo de la distribución de Linux que se utilizó. El comando ”Yum” para distribuciones basadas en Red Hat y el comando “apt-get” para distribuciones basadas en Debian. Se pueden programar las actualizaciones automáticamente para que se ejecuten un día y hora especifica en la semana, se podrán recibir notificaciones por correo electrónico cuando existan actualizaciones críticas para el Sistema Operativo. Muchas de estas actualizaciones abordan las vulnerabilidades de seguridad encontradas por la comunidad de Linux, por lo que mantener los sistemas operativos lo más actualizados posible es esencial.

Manteniendo nuestro Sistema Operativo con las ultimas actualizaciones y parches de seguridad, no solo estamos mejorando funcionalidades y rendimiento del Sistema Operativo, también corrigen bugs y vulnerabilidades para que no sean explotadas por ningún atacante.

G. Password Aging

Hay un comando que permite cambiar el número de días entre cambios de contraseñas y la fecha del último cambio de contraseña. Este comando es el change. Básicamente, lo que se logra con este comando es determinar cuando un usuario debería cambiar su contraseña. Es decir, cuanto tiempo tiene de vigencia determinada contraseña establecida por el usuario en cuestión: Este comando nos permite deshabilitar el Password Aging:

```
chage -M 99999 userName
```

Fig. 9: Password Aging.

Si necesitas obtener información respecto a la expiración de contraseña de un usuario determinado, escribe lo siguiente:

```
chage -l userName
```

Fig. 10: Password User Information

El siguiente comando, te permite cambiar varios atributos relacionados a la gestión del cambio de las contraseñas:

```
chage -M 60 -m 7 -W 7 userName
```

Fig. 11: Password Change

Hagamos un pequeño desglose de este comando y sus parámetros de ejemplo:

- -M: Se refiere a la máxima cantidad de días en que esa contraseña es válida.
- -m: Se refiere a la mínima cantidad de días que requieren que pasen entre eventos de cambios de contraseña.
- -W: La cantidad de días con las que se notifica al usuario respecto a la necesidad de cambiar las contraseñas.
- nombreUsuario: el nombre de usuario Linux que estamos gestionando.

Como regla general, debe nunca reutilice contraseñas antiguas. Puede restringir fácilmente a los usuarios el uso de sus contraseñas antiguas en la misma máquina.

Se debe modificar el archivo de contraseñas anterior el cual es: /etc/security/opasswd . Esto solo se puede cambiar usando el módulo PAM en Linux.

Para RHEL (Red Hat Enterprise Linux), CentOS y Fedora modifique el archivo “/etc/pam.d/system-auth”

Para Debian y Ubuntu modifique el archivo “/etc/pam.d/common-password”.

En la sección “auth” se debe agregar la siguiente línea: “auth enough pam_unix.so likeauth nullok”

Si se desea permitir que un usuario reutilice su contraseña de una serie de contraseñas que se usaron por última vez, se agrega la siguiente línea en la sección “contraseña” “contraseña suficiente pam_unix.so nullok use_authtok md5 shadow recordar = 3”. Después de realizar estos procedimientos, mostrar un error al usuario que este intentando utilizar una de las 3 ultimas contraseñas utilizadas en este servidor. [3]

H. Encriptación Disco Duro

Hoy en día las distribuciones Linux más usadas permite encriptar los discos duros antes de la instalación. Con el fin de proteger la información almacenada en caso de pérdida el dispositivo. La persona que intente acceder a la información conectando el disco duro en otro equipo no va a poder tener acceso a los archivos almacenados porque no tiene la llave para desencriptar el disco duro.

Para los sistemas operativos que no permite realizar la encripción del disco duro antes de la instalación, existen muchos paquetes que nos permite realiza la encripción de los

discos duros. Uno de los paquetes más populares es dmcrypt . la cual se podría instalar en cualquier distribución Linux con los siguientes comandos:

- yum install cryptsetup-luks - Sistemas operativos Red Hat/Fedora.
- apt-get install cryptsetup - Sistemas operativos basados en Debian

Después de la instalación se puede ejecutar el proceso de encripción de la partición que deseemos ejecutando el siguiente comando c¸ryptsetup -y -v luksFormat (partición)".[4]

I. Antivirus

Los servidores o equipos de cómputo instalados con cualquier distribución Linux son menos vulnerables a la propagación de virus informáticos. Pero se debe mantener el sistema 100% seguro por lo que algunos antivirus que se podrían instalar en nuestro equipo o Servidor son los siguientes:

- Sophos: Es una de las pocas empresas que desarrollan Software para Linux, cuenta con un sistema de monitoreo en tiempo real, análisis por demanda, es fácil de instalar y configurar. Su directorio de Malware no es de código abierto.
- Eset: Uno de los mejores antivirus para los sistemas operativos Windows. Es un antivirus sencillo y fácil de utilizar, peor muy eficaz a la hora de detectar amenazas. Cuenta con uno de los sistemas más precisos para detectar virus y malware y spyware, además de permitiros proteger nuestra red. Se actualice automáticamente.
- Clamav: Un antivirus de código abierto al igual que Linux, por lo que la distribución de Antivirus es actualizada constantemente por los desarrolladores alrededor del mundo; La administración de Clamav es por línea de comando por lo que se debe tener conocimientos avanzados en Linux para su correcta configuración y funcionamiento.

Podemos instalar el paquete clamav con el siguiente comando:

- yum install clamav - Sistemas operativos Red Hat/Fedora.
- apt-get install clamav - Sistemas operativos basados en Debian.

Después de instalado el antivirus debemos actualizar la base de datos, actualización que se realizara en segundo plano, debemos detener el servicio con el comando: systemctl stop clamav-freshclam
Actualizamos la base de datos:

- Freshclam

Se inicia nuevamente el servicio:

- systemctl start clamav-freshclam

Se deben programar escaneos automáticos a ficheros y directorios del servidor, la sintaxis es la siguiente:

- clamscan -r /path/to/directory

Adicionalmente se puede configurar el paquete clamav-milter para filtrar los correos electrónicos de sendmail, la cual se encargará de escanear todo el correo electrónico entrante, también se puede configurar para la filtración del correo Spam,

J. VPN

Una VPN, de la sigla en inglés asociada a Red Privada Virtual, es una de las formas de crear conexiones seguras entre computadores remotos, presentándose como si éstos se encontraran en una red privada local. Esto permite configurar a los servicios como si estuviesen en una red privada, así como de conectar servidores de manera segura.

Las redes privadas son las redes que se encuentran habilitadas únicamente para ciertos usuarios o servidores. En algunas organizaciones, la red privada está disponible en algunas regiones con la misma amplitud que la red del Centro de Datos.

Una VPN brinda diferentes beneficios. Algunos de estos se destacan a continuación:

- Seguridad: una VPN proporciona una protección más sólida ya que todos los datos están encriptados. Esto otorga seguridad adicional, algo que no proveen los firewalls.
- Acceso remoto: para garantizar una seguridad alta, muchas organizaciones permiten el acceso remoto solo a través de su VPN.
- Cifrado de ISP: ISP significa proveedor de servicios de Internet. Si utilizas una conexión WiFi-pública, el ISP puede leer todos tus datos no cifrados. Al usar una VPN, puedes mantener tus datos seguros y encriptados al no permitir que ni siquiera el ISP los lea.
- Anonimato: una VPN permite a los usuarios mantener el anonimato mientras navegan por Internet. Las IP no son rastreables.
- Cambio de IP: Configurar una VPN permite a los usuarios cambiar sus IP y navegar de forma segura. Esto, en ciertos casos, se usa en regiones que tienen restricciones basadas en la ubicación.
- Desbloquea sitios web: ciertos sitios web están bloqueados en algunas regiones geográficas. Una VPN mantiene el anonimato y, por lo tanto, se usa comúnmente para evitar la censura de Internet y desbloquear sitios web.
- Regulación: ciertos ISP reducen el ancho de banda del usuario en función del contenido. Tal estrangulamiento puede evitarse utilizando una VPN.

Para realizar la configuración de un servidor VPN en Linux se debe instalar el siguiente paquete:

- yum install net-tools - Sistemas operativos Red Hat/Fedora.
- apt-get install net-tools - Sistemas operativos basados en Debian.

K. Servidor de archivos NFS

NFS (Network File System - Sistema de Archivos en Red) es el sistema nativo utilizado por Linux para compartir carpetas en una red. Mediante NFS, un servidor puede compartir sus carpetas en la red. Desde los PCs de los usuarios se puede acceder a dichas carpetas compartidas y el resultado es el mismo que si estuvieran en su propio disco duro.

La configuración NFS se debe realizar tanto en el servidor como en los equipos remotos. La instalación y configuración en el servidor es la siguiente:

- yum -y install nfs-utils - Sistemas operativos Red Hat/Fedora.
- yum -y install rpcbind - Sistemas operativos Red Hat/Fedora.
- apt-get install nfs-kernel-server - Sistemas operativos basados en Debian.
- apt-get install nfs-common - Sistemas operativos basados en Debian.

Iniciar el servicio del servidor NFS y habilitarlo para el inicio del sistema (CentOS):

- systemctl start rpcbind nfs-server
- systemctl enable rpcbind nfs-server

Se edita el archivo /etc/exports y se configura el archivo compartido de la siguiente manera: /path 192.168.1.0/24 (rw)

Después de adicionar los recursos compartidos se debe ejecutar el comando “exports -a”, las opciones para compartir los recursos en NFS son las siguientes:

- ro : lectura
- rw : lectura y escritura
- link_relative : convierte los enlaces simbólicos absolutos en enlaces simbólicos relativos.
- link_squash : trata las consultas como si vinieran del usuario nobody (lo más seguro). O sea, que evitamos que los usuarios conectados actúen como root.
- no_root_sqash : Desactiva lo anterior, y por tanto implica algo de riesgo porque habilita los privilegios de root.
- all_squash : Realiza la acción roo_squash para todos los usuarios, incluido los usuarios root.
- no_all_squash : Habilita la autorización del usuario.

La instalación NFS en el cliente es la siguiente:

- yum install nfs-utils - Sistemas operativos Red Hat/Fedora.
- yum install rpcbind - Sistemas operativos Red Hat/Fedora.
- apt-get install nfs-common - Sistemas operativos basados en Debían.

Para realizar un montaje de red permanente se edita el archivo “/etc/fstab” y adicionamos la ruta que deseamos añadir a nuestro sistema de archivos de la siguiente manera:

- 192.168.0.10:/path_servidor /carpeta_compartida nfs defaults 0 0

Después de añadir esta línea en el archivo se debe reiniciar el equipo para que al iniciar monte la carpeta sin necesidad de realizarlo manualmente.: mount -t nfs 192.168.8.114:/compartido /compartido-centos/

Si se quiere definir qué hosts pueden y cuales no pueden utilizar los recursos compartidos por NFS, se editan los siguientes archivos adicionando las direcciones ips o segmentos de red a los cuales se les permite o restringe el acceso:

- hosts.deny
- hosts.allow

L. SAMBA

Para poder compartir directorios u otros recursos desde un sistema operativo Linux a maquinas con sistemas operativos Windows, se puede utilizar Samba. La cual nos permite que equipos Windows accedan a directorios o ficheros los cuales están almacenados en servidores con sistema operativo Linux, este protocolo es muy utilizado a nivel organizacional en pequeñas empresas, que almacenan sus datos en servidores locales.

La instalación del servidor Samba es la siguiente:

- yum install samba - Sistemas operativos Red Hat/Fedora.
- apt-get install samba - Sistemas operativos basados en Debian.

Se debe editar el archivo “ /etc/samba/smb.conf” y al final del archivo se agrega el siguiente bloque de líneas:

```
[share]
comment = Ubuntu File Server Share
path = /srv/samba/share
browsable = yes
guest ok = yes
read only = no
create mask = 0755
```

Los parámetros que agregamos son los siguientes:

- comment: es una breve descripción de la carpeta que se quiere compartir.
- path: la ruta del directorio que se va a compartir.
- Browsable: permite a los clientes de Windows examinar el directorio compartido utilizando el Explorador de Windows.
- Guest ok: permite a los clientes conectarse al recurso compartido sin proporcionar una contraseña.
- read only: determina si el recurso compartido es de sólo lectura o si se conceden privilegios de escritura. Los privilegios de escritura sólo se permiten cuando el valor es no, como se ve en este ejemplo. Si el valor es sí, el acceso al recurso compartido es sólo lectura.
- create mask: determina los permisos que tendrán los nuevos archivos cuando se creen.

Se debe reiniciar el servicio Samba para que tome los cambios.

M. Habilitar SUDO

SUDO es una herramienta de los sistemas operativos Linux, la cual permite a los usuarios realizar la ejecución de comandos como super-usuario u otro usuario de acuerdo con las especificaciones en el archivo /etc/sudoers.

Por seguridad el comando SUDO requiere que los usuarios se autentiquen primero a sí mismos. Es decir, con su propia clave de acceso, El comando Sudo es necesario ejecutarlo desde una terminal de trabajo en Linux.

Por defecto el paquete SUDO viene preinstalado en varias distribuciones Linux, en caso de ser necesario la instalación del paquete, se podrá instalar con los siguientes comandos:

- yum install sudo - Sistemas operativos Red Hat/Fedora.

- apt-get install sudo - Sistemas operativos basados en Debian.

El archivo /etc/sudoers maneja la siguiente sintaxis para su configuración y manejo de permisos.

- root ALL=(ALL:ALL) ALL : Esta línea nos indica que el usuario root tiene privilegios ilimitados y puede ejecutar cualquier comando del sistema.
- % admin ALL=(ALL) ALL : El signo "%" especifica que es un grupo . Cualquier usuario que pertenece a este grupo de administradores que tiene los mismos privilegios que el usuario root.
- %sudo ALL=(ALL) ALL : Todos los usuarios del grupo sudo tienen privilegios para ejecutar cualquier comando.

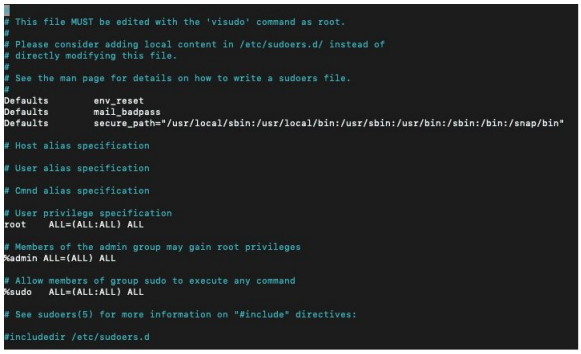

Fig. 12: Contenido del archivo /etc/sudoers

La configuración del archivo sudo es necesario en un servidor donde se conectan varios usuarios. No compartir la contraseña root es una medida de seguridad necesaria, adicional que cada persona trabaje con su propio usuario deja registro de las actividades que realizan los diferentes usuarios sobre el sistema operativo. [5]

N. Copias de respaldo

Prevenir la perdida de datos o información en cualquier sistema operativo, bases de datos o carpeta compartida es una tarea esencial de un administrador de cualquier infraestructura tecnológica. Se deben tener políticas establecidas de copias de respaldo, como que información se va a respaldar, con que periodicidad se va a respaldar y en donde se almacenara esta información.

Los sistemas operativos Linux permiten el manejo de varias herramientas para el manejo de las copias de seguridad. Se debe tener conocimientos previos en la creación de scripts para configurar algunas de ellas. Las herramientas más utilizadas ya que son de fácil implementación son las siguientes:

- Rsync es una de las herramientas más usadas en Linux para realizar backups. Permite hacer copias incrementales, tanto local como remotamente. Rsync tiene la capacidad de actualizar sistemas de archivos completos; preservando los enlaces, dueños, permisos y privilegios. Rsync es una herramienta de línea de comandos, aunque hay varios front ends disponibles (como Grsync) pero estos anulan la flexibilidad de tener una simple herramienta de línea de comandos para hacer backups. Podemos crear scripts simples para usar, en conjunto con cron, puedes crear backups automatizados. El paquete rsync esta disponible en todas las distribuciones Linux, y se puede complementar con los archivos con extensión .tar para su comprensión y almacenamiento.
  - yum install rsync - Sistemas operativos Red Hat/Fedora.
  - apt-get install rsync - Sistemas operativos basados en Debian.
- Bacula es una herramienta de copia de seguridad de código abierto de nivel empresarial. Su funcionamiento y configuración es un poco compleja. Es un sistema cliente-servidor que puede ejecutarse en un sistema doméstico pequeño o en una implementación de respaldo de una gran empresa. Su tarea no requiere la intervención de ningún operador informático o administrador del sistema. Bacula utiliza una base de datos como back end para almacenamiento, por defecto viene configurada con Postgresql , pero también soporta MySQL y Maria DB. Para su instalación se debe realizar un registro previo en el siguiente enlace: https://blog.bacula.org/bacula-binary-package-download/ Después se ejecutan los siguientes comandos en una terminal:
  - wget -c https://www.bacula.lat/wp-content/uploads/2018/09/bacul -O /usr/local/bin/_bacula_community_install.sh
  - chmod a+ /usr/local/bin/_bacula_community_install.sh
  - /usr/local/bin/_bacula_community_install.sh
- Amanda permite a un administrador hacer un backup individual a un servidor y también a múltiples hosts. Es robusto, confiable, y flexible. Amanda usa dumps nativos de Linux y/o archivos con extensión tar para facilitar el proceso de hacer copias de seguridad. También puede usar Samba para hacer backups a los clientes Windows. La configuración de Amanda se maneja en 2 aplicaciones una para el servidor y otra aplicación para el cliente. Los paquetes para instalar son:
  Server
  - yum install amanda-server - Sistemas operativos Red Hat/Fedora.
  - apt-get install amanda-server - Sistemas operativos basados en Debian.

  Client
  - yum install amanda-client - Sistemas operativos Red Hat/Fedora.
  - apt-get install amanda-client - Sistemas operativos basados en Debian.

O. Manejo de Logs

Conocer donde se encuentran los registros del sistema operativo nos pueden ayudar a conocer más del funcionamiento de este y poder solucionar vulnerabilidades las cuales no se pueden encontrar a simple vista por el administrador del sistema. Los logs se encargan de almacenar

los mensajes generados por los programas, aplicaciones y demonios y enviarlos a un destino predefinido. Las distribuciones Linux almacenan los registros del sistema en la ruta /var/log/, algunos de los logs que un administrador de servidores Linux debe conocer son los siguientes:

- /var/log/messages - Log General del sistema Linux.
- /var/log/auth.log - Log donde lista todos los intentos de autenticación al sistema operativo (Sistemas basados en Debian).
- /var/log/secure - Log donde lista todos los intentos de autenticación al sistema operativo (Sistemas basados en Red Hat y Fedora).
- /var/log/utmp - Log donde registra los usuarios conectados en un periodo de tiempo en el sistema operativo.
- /var/log/wtmp - Log donde registra todos los usuarios que se han logueado y deslogueado del Sistema Operativo.
- /var/log/kern.log - Log donde se almacena información del Kernel del Sistema Operativo.
- /var/log/boot.log - Log donde que contiene información y mensajes de los inicios del Sistema Operativo.

P. Particionamiento Disco Duro

Crear particiones al momento de instalar el Sistema Operativo permite minimizar el impacto de perdida de información en caso de que se presente algún daño en el sistema o Disco duro, se perderá la información de la partición que presenta el error si es que no se pudiera reparar. Realizar el particionamiento da un mejor rendimiento y seguridad adicional mantendrá la información más organizada aprovechando el tamaño del disco duro al máximo. El esquema de particionamiento recomendado es el siguiente:

- / la barra / sola representa la raíz del árbol del filesystem.
- /boot Contiene todos los archivos que son necesarios para el proceso de arranque.
- swap La partición de intercambio es donde se extiende la memoria del sistema al dedicar parte del disco duro a esta.
- /usr Contiene los archivos ejecutables y los recursos compartidos que no son críticos del sistema.
- /home Contiene los archivos ejecutables y los recursos compartidos que no son críticos del sistema.
- /tmp Este es un lugar para archivos temporales. Los tmpfs que se montan en él o en los scripts durante el inicio, generalmente, eliminan esto en el arranque.
- /var Esto significa variable y es un lugar para los archivos que están en un estado cambiante. Por ejemplo, el tamaño que aumenta y disminuye.
- /opt Contiene software de complemento, los programas más grandes se pueden instalar aquí en vez de la partición /usr .

VI. CONCLUSIONES

- La instalación correcta del Sistema Operativo es el primer paso para garantizar el correcto funcionamiento del servidor o equipo de cómputo y que tan seguro pueda ser.
- Asegurar cualquier sistema operativo es una tarea que debe ser realizada por una persona experta, que conozca el funcionamiento del sistema operativo, que debe asegurar.
- Una distribución Linux puede ser tan segura como el administrador lo desee, aplicando los parches de seguridad correctos de acuerdo con los paquetes instalados, medidas de control de acceso eficientes, con el fin de mitigar el acceso no autorizado.
- Se debe tener claro el servicio que va a prestar el servidor, para instalar únicamente los paquetes necesarios con el fin de minimizar la cantidad de puestos abiertos y servicios ejecutándose en el servidor.
- Se deben tener políticas de seguridad claras, con el fin de que sean cumplidas por todo el personal que accede o utiliza un servidor sin importar la actividad que realice.

REFERENCES

[1] Remo Suppi Boldrito and Josep Jorba Esteve," Administración avanzada del sistema operativo GNU/Linux" Pag 14, GNUFDL PID_00212464
[2] James Turnbull Hardering Linux Pag 60 Edt. Apress
[3] James Turnbull Hardering Linux Pag 238 Edt. Apress
[4] Donald A. Tevault, Mastering Linux Security and Hardering Pag 22 Edt. PACK
[5] Donald A. Tevault, Mastering Linux Security and Hardeing Pag 247 Edt. PACK