Linux 100%

OpenWebinars

ÍNDICE

USUARIOS, GRUPOS Y PERMISOS

- Para añadir una nueva cuenta de usuario en Linux puedes utilizar el comando adduser, con un sudo o su delante, dado que la acción requiere de permisos de super usuario.

Sudo adduser usuario

- Si para añadir una cuenta utilizamos adduser, para eliminar un usuario el comando a utilizar es userdel.

Sudo user del usuario

- Para cambiar el password de una cuenta en concreto. Para ello se usa el comando passwd.

Sudo passwd usuario

- Para ver todos los usuarios del Sistema debemos acceder al fichero/ etc/ passwd, tener en cuenta que este fichero está considerado como crítico.

Cat / etc / passwd

- Para añadir un grupo, basta con abrir nuevamente una ventana de terminal.

Sudo groupadd grupo

- Al igual que en el caso de los usuarios, recomendamos NO borrar en ningún caso uno de los grupos creados por el sistema en la instalación inicial.

Sudo del user usuario grupo

- Para todos los grupos del sistema.

Gedit /etc/ group

- Cambiar el propietario de un fichero.

- A grandes rasgos, en Linux todos los archivos o directorios pertenecen a un usuario y un grupo.

CAMBIAR USUARIO PROPIETARIO CON CHOWN

- El comando que sirve para cambiar el propietario de un determinado archivo o directorio a nivel de user.

Sudo chown usuario ruta-fichero:

- En caso de que se trate de un directorio.

Sudo chown -R usuario ruta-directorio

- Cambiar grupo propietario con Chgrp.

Sudo chgrp grupo ruta-archivo

Sudo chgrp -R grupo ruta-directorio

- Asignar permisos de lectura, escritura y ejecución de un fichero.

Sudo chmod XYZ ruta-archivo

- En esta representación, X se refiere a los permisos del usuario propietario, Y a los del grupo propietario y Z

0. Ningún permiso
1. Ejecución
2. Escritura
3. Escritura y ejecución
4. Lectura
5. Lectura y ejecución
6. Lectura y escritura
7. Lectura, escritura y ejecución

USAR CHMOD EN MODO TEXTO

Sudo chmod [u,g,o][=,+,-][r,w,x] ruta archivo

u: Usuario.

g: Grupo.

o: Otros.

ACCESOS REMOTOS

- Secure shell ssh, al igual que telnet se utiliza para gestionar el login de acceso a un sistema remoto. Pero a diferencia de telnet, ssh establece una conexión cifrada con el sistema remoto, además de proporcionar otras posibilidades.

Ssh -l nusuarior -p puertot maquinaremota

- **-l nusuario:** usuario autorizado para conectarse al servidor sshd remoto.
- **-p puerto:** puerto por el que se escucha el servidor SSH de la máquina remota (normalmente es el puerto 22).
- **Maquinaremota:** es la dirección IP o un nombre válido convertible a dirección IP mediante el uso de resolución de nombres (DNS).

SSH. Túneles

- A pesar de la utilidad mencionada anteriormente, la funcionalidad de SSH no termina aquí, sino que nos brinda la posibilidad de crear túneles cifrados entre dos computadoras.

- La orden que se utiliza para la creación de estos túneles es:
  ssh nusuario@maquinaremota -p puerto -L plocal:servidorremoto:premoto -N

Descripción

- **-p puerto:** puerto por el que escucha el demonio sshd de la máquina remota.
- **-N:** indica que no se debe iniciar una terminal tras la conexión.
- **-L:** suministra los parámetros para el túnel.
- **Plocal:** puerto en la máquina local encaminado al puerto servidorremoto.

- **Servidorremoto:** es la dirección de máquina que corre el servicio a contactar, desde el punto de vista de la máquina que corre el servidor sshd (maquinaremota).
- **Premoto:** es el puerto del servicio que se desea contactar en la máquina servidorremoto.
- **nusuario:** es el usuario autorizado para conectarse al servidor sshd.
- **Maquinaremota:** es el servidor ssh que hace de puente entre la máquina local y el servidor servidorremoto.

VNC vía túnel SSH

- Muchas veces la posibilidad de ejecutar programas gráficos mediante el redireccionamiento de X resulta insuficiente y se requiere acceder a un escritorio remoto por completo.

- La solución radica en instalar un servidor VNC en la máquina remota y acceder desde la máquina local con el programa cliente adecuado.

- Una vez instalado tightvncserver en la máquina remota se debe ejecutar (como usuario no privilegiado) la orden: $vncserver

- Se solicitará una contraseña de acceso y otra para "modo observar". Inmediatamente se crea un nuevo servidor X y se le signa el número correspondiente, por ejemplo: 1 En la máquina local.

- Se ejecuta vinagre desde una terminal o con "Ejecutar comando" (Alt+F2)con los siguientes parámetros:

  vinagre maquinaremota:n

Descripción

- **Maquinaremota:** equipo en que se encuentra ejecutando el servidor VNC.
- :n:Identificador del servidor X lanzado por el servidor VNC.

  vinagre servidor.mindominio.org:1

SSH. Acceso automático mediante clave pública

- El empleo de contraseñas supone la intervención del operador y también impone que las órdenes se puedan ejecutar sin recurrir a un modo interactivo.
- **ssh-keygen -t rsa**
- Generating public/private dsa key pair.

- Pulsamos Enter cuando nos solicita una frase de contraseña (passphrase)
- El archivo id_rsa contiene nuestra clave provada y id_rsa.pub nuestra clave pública. Ahora queda enviar la clave pública a la máquina remota para que no sea necesario el uso de la contraseña.
- **$ ssh-copy-id -i archivo nusuario@maquinaremota**
- o, por ejemplo: **$ ssh-copy-id -i ~/.ssh/id_rsa.pub prueba@servidor.midominio.org**

SQUID: SERVIDOR PROXY-CACHÉ

- Es un servidor situado entre la máquina del usuario y otra red (a menudo internet)

- Actúa como protección separando las dos redes y como zona caché para acelerar el acceso a páginas web o poder restringir el acceso a contenidos.

Funciones

- Permite el acceso web a máquinas privadas (IP privada) que no están conectadas directamente a Internet.
- Controla el acceso web aplicando reglas.
- Registra el tráfico web desde la red local hacia el exterior.
- Controla el contenido web visitado y descargado.

- Controla la seguridad de la red local ante posibles ataques, intrusiones en el sistema, etc.
- Funciona como un caché de páginas web. Es decir, almacena las páginas web visitadas por los usuarios y de esta manera las puede enviar a otros usuarios sin tener que acceder a Internet de nuevo.
- Guarda en caché las peticiones DNS e implementa una caché para las conexiones fallidas.
- Registra logs de todas las peticiones.

Ventajas

- Reduce los tiempos de respuesta.
- Si la página web que se solicita está en la caché del servidor, ésta se sirve sin necesidad de acceder de nuevo al servidor original, con lo cual se ahorra tiempo.
- Disminuye el tráfico en la red y el consumo de ancho de banda.
- Cortafuegos.

- Si la página web está almacenada en la caché del servidor, la petición no sale de la red local y no será necesario hacer uso de la línea exterior consiguiendo así un ahorro en la utilización del ancho de banda.
- Cuando se utiliza un servidor proxy-caché, éste comunica con el exterior, y puede funcionar como cortafuegos, lo cual aumentará la seguridad del usuario respecto a la información a la que se acceda.
- Filtrado de servicios.

- Es posible configurar el servidor proxy-caché dejando sólo disponibles aquellos servicios (HTTP, FTP...) que se consideren necesarios, impidiendo la utilización del restones cursadas.
- Soporta el protocolo ICP que permite integrar cachés que colaboran y permite crear jerarquías de cachés y el intercambio de datos.

INSTALACIÓN

- Squid dispone de una sencilla instalación, mediante el comando apt-get.
- Los componentes de Squid se instalan en los siguientes directorios:
    - Archivos/Directorios.
    - Descripción:
        - /usr/sbin/
    - Directorio del ejecutable:
        - /var/run/

- Archivo con el PID del proceso:
  - /var/log/squid/
- Directorio de logs. ara relacionado con la directiva access_log:
  - /var/spool/squid
- Directorio caché. Relacionado con el cache_dir:
  - /etc/squid/
- Archivos de configuración:
  - /usr/lib/squid/

- Complementos:
    - /etc/rc.d/
- Scripts de arranque:
    - /usr/share/dos/squid/
- Documentación.

CONFIGURACIÓN BÁSICA

- En el servidor habrá que:
  - Determinar el espacio en disco dedicado al servidor proxy-caché.
  - Configurar el propio servidor a nivel de puerto, directorios, usuarios, etc.
  - Arrancar el servicio.
- En el cliente para que utilice el servidor proxy habrá que realizar una:
  - Configuración manual (servidor, protocolos y puerto)
  - Configuración automática utilizando un archivo proxy.pac.
- acl permitidos src /etc/squid/permitidos

- Una configuración básica bajo estas condiciones sería:
  - acl todo src 0.0.0.0/0.0.0.0
  - acl permitidos src/etc/squid/permitidos
  - acl web_denegadas dstdomain .chicas.com .sex.com
  - acl horario time MTWHF 9:00-17:00
  - acl mañana time 9:00-14:00

- acl url_denegar url_regex/etc/squid/denegar.txt
- http_access allow permitidos horario
- http_access deny web_denegadas
- http_access deny url_denegar
- http_access deny todo

IPTABLES

- Es una utilidad de línea de órdenes para configurar el cortafuegos del kernel de Linux implementado como parte del proyecto Netfilter.
- El término iptables también se usa comúnmente para referirse a dicho cortafuegos del kernel.
- Puede configurarse directamente con iptables, o usando uno de los muchos froten existentes de consol y gráficos.
- El término iptables se usa para IPv4, y el término ip6tables para IPv6.

- Tanto iptables como ip6tables tienen la misma sintaxis, pero algunas opciones son específicas de IPv4 o de IPv6.
- Para trabajar con iptables es necesario tener permisos administrativos, por lo que deberemos usar sudo.
- No hay un límite respecto de cuán anidadas pueden estar las cadenas.
- Hay tres cadenas básicas /INPUT,OUTPUT y FORWARD: ENTRADA, SALIDA, y REENVÍO) y el usuario puede crear tantas como desee.

- Una regla puede ser simplemente un puntero a una cadena.
- -=SYNTAXIS=-

  iptables [tabla] [COMANDOS] reglas DESTINO

1. Filter table (tabla de filtros). Esta tabla es por la cual pasan todos los paquetes sin distinción y es la responsable del filtrado. Contiene las siguientes cadenas:

a) **INPUT**: los paquetes que sean destinados al sistema atraviesan esta cadena.

b) **OUTPUT**: todos los paquetes que han sido creados por el sistema pasan por esta cadena.

c) **FORWARD**: todos los paquetes que simplemente pasan por el sistema para ser encaminados a su destino.

2. Nat table (tabla de traducción de direcciones de red).

- Esta tabla tiene a su encargo configurar las reglas de escritura de direcciones o de los puertos de los paquetes.
- El primer paquete que entre al sistema de cualquier conexión pasa a través de esta tabla; los veredictos determinan cómo van a reescribirse todos los paquetes de la conexión.
- Contiene las siguientes cadenas redefinidas:

a) PREROUTING chain (Cadena de PRERUTEO): los paquetes con revisados en esta regla antes de que sea consultada del ruteo local, principalmente el DNT (destination-NAT).

b) POSTROUTING chain (Cadena de POSTRUTEO): lso paquetes al salir pasan por esta cadena después de tomar la decisión del ruteo, principalmente el SNT (source-NAT).

c) OUTPUT chain (Cadena de SALIDA): permite hacer un DNAT solamente en los paquetes generados.

3. Mangle table (Tabla de destrozo): esta tabla ajusta las opciones de los paquetes. Todos los paquetes pasan por esta table. Está diseñada para fines avanzados, por eso todas las cadenas están pre definidas.

a) **PREROUTING chain** (Cadena de PRERUTEO): Todos los paquetes que logran entrar a este sistema, antes de que el ruteo decida si el paquete debe ser reenviado (cadena de REENVÍO) o si tiene destino local (Cadena de ENTRADA).

b) INPUT chain (Cadena de ENTRADA): Todos los paquetes destinados para este sistema pasan a través de esta cadena.

c)FORWARD chain (Cadena de REDIRECCIÓN): Todos los paquetes que exactamente pasan por este sistema pasan a través de esta cadena.

d) OUTPUT chain (Cadena de SALIDA): Todos los paquetes creados en este sistema pasan a través de esta cadena.

e) POSTROUTING chain (Cadena de POSTRUTEO): Todos los paquetes que abandonan este sistema a través de esta cadena.

Además de las cadenas ya incorporadas, el usuario puede crear todas las cadenas definidas por el usuario que quiera dentro de cada tabla, las cuales permiten agrupar las reglas en forma lógica.

Comandos

- **Iptables -A:** append, añadir regla.
- **Iptables -D:** borrar una regla.
- **Iptables -F:** elimina todas las reglas.
- **Iptables -P:** modifica las políticas para una cadena (input, forward, output).
- **Iptables -L:** lista todas las reglas.
- **Iptables -N:** crea una cadena de un usuario.
- **Iptables -R:** reemplaza una regla.
- **Iptables -X:** elimina una cadena definida por un usuario.

Reglas

- **i:** interfaz de entrada res por donde entrará el paquete. Solo funciona para (INPUT, FORWARD y PREROUTING) (eth0, eth1, ppp0...).
- **o:** el interfaz de salida de re por donde saldrá el paquete (eth0, eth1, ppp0,...).
- **s 0.0.0.0/0:** compara los paquetes que vienen de la dirección de acceso (cualquiera en este caso).

- **d 0.0.0.0/0:** compara los paquetes que salen de la dirección de origen, La IP puede sustituirse por un hostname tanto en -s como en -d.
- **p TCP:** tipo de puerto o protocolo que será comparado con la regla (TCP, UDP,ICMP).
- **Sport:** puerto de origen. Puede abrir un rango usando #.# (ejemplo. 240:241)
- **Dport:** puerto de destino.
- **M:** define que se aplica la regla si hay una coincidencia específica.

Destino

- **J:** Destino de la regla, (ACCEPT, DROP, LOG, REJECT, QUEUE, DNAT o SNAT).
- **ACCEPT:** acepta la conexión.
- **DROP:** deniega el acceso.
- **REJECT:** rechaza la conexión y reenvía el paquete a su origen.
- **QUEUE:** envía el paquete hacia las reglas del usuario.

- **LOG:** todos los paquetes que coincidan por esta regla se guardan en un log.

- **SNAT:** un estado virtual donde difiere si la dirección fuente original difiere del envío destinado.

- **DNAT:** un estado virtual donde coincide si el destino difiere del lugar donde son reenviados.

Limpieza de las reglas

- Eliminamos todas las reglas existentes. #Iptables -F
- Eliminamos las cadenas definidas por usuarios. #Iptables -X
- Llenamos a cero todos los paquetes y contadores de todas las cadenas. #Iptables -Z
- Eliminamos las reglas en la tabla NAT. #Iptables -t nat -F

- Nota: averiguar sobre esta regla.
- Eliminamos las reglas de la tabla manle #Iptables -t mangle -F
- Nota: averiguar sobre esta regla.

Establecer las políticas por defecto (accept o drop)

- Denegamos todos las entradas.
  - iptables -P INPUT DROP
- Denegamos todas las salidas.
  - iptables -P OUTPUT DROP
- Denegamos todos los reenvios de paquetes.
  - iptables -P FORWARD DROP

Listamos todas las reglas actuales que contiene nuestro IPTABLES:

[root@localhost~]# iptables -L

- Chain INPUT (policy DROP): target prot opt source destination
- Chain FORWARD (policy DROP): target prot opt source destination
- Chain OUTPUT (policy DROP): target prot opt source destination

Insertamos nuestras reglas

- Aceptar todas las peticiones locales:

  Iptables -A INPUT -i lo -j ACCEPT

  Iptables -A OUTPUT -o lo -j ACCEPT

- A nuestra ip acceso hacia fuera y hacia dentro total:

  Iptables -A INPUT -s 172.12.1.2 -p tcp --dport 22 -j ACCEPT

  Iptables -A OUTPUT -d 172.16.100.94 -j ACCEPT

- Solo aceptamos SSH de este cliente:

  Iptables -A INPUT -s 172.16.1.2 -p tcp --dport 22 -j ACCEPT

  Iptables -A OUTPUT -d 172.16.1.2 -p tcp --sport22 -l ACCEPT

- Aceptamos todo el tráfico de este cliente. Aquí se aceptan los pings:

  Iptables -A INPUT -s 172.16.1.2 -j ACCEPT

  Iptables -A OUTPUT -d 172.16.1.2 -j ACCEPT

- Permitir resolver nombres (DNS). Si hacemos ping nos dirá que no está permitido, entonces hacemos el siguiente paso:

  Iptables -A INPUT -i eth0 -p udp --sport 53 -j ACCEPT

  Iptables -A OUTPUT -O eth0 -p udp --dport 53 -j ACCEPT

- Aceptar las peticiones del ping que salen del servidor:

  Iptables -A INPUT -p icmp -m state --state ESTABLISHED, RELATED -j ACCEPT

  Iptables -A OUTPUT -P ICMP -J ACCEPT

- También podemos permitir que nos den PING desde cualquier lugar:

  Iptables -A INPUT -i eth0 -p ICMP -j ACCEPT

- Abrimos el puerto 80 para un servidor web:

  Iptables -A INPUT -p tcp --sport 80 -j ACCEPT

  Iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT

RETO: Instalación, configuración de un proxy-caché

En el presente reto, se insta al alumno a realizar una instalación del proxy-caché squid, así como el establecimiento de algunas reglas de bloqueo.

- Instalar squid bajo ubuntu.
- Configurar squid como proxy-caché.
- Configurar reglas para una determinada IP que bloquee el acceso a facebook y twitter.