Practica-2-Leccion-1-Resuelta.pdf

Zukii

Ingeniería de Servidores

3º Doble Grado en Ingeniería Informática y Administración y Dirección de Empresas

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

Práctica 2 – Lección 1 - Resuelto

Apuntes clase:
(Tanto CentOS como Ubuntu).

1. Instalarlo (sshd)
2. Deshabilitar el acceso de root -> Por seguridad. Si quisieramos, entrar como usuario normal y sudo.
3. Cambiar puerto de ssh. (Por defecto 22 a otro). Por practicar. OJO Firewall
4. Acceso por ssh sin contraseña. (Conectarme entre máquinas con y sin contraseña. OJO MV no pueden entrar al anfitrión pero si al revés).

Llave pública que la conozca todo el mundo pero la privada solo nosotros. Puedo cifrar con pública y descifrar con privada.

putty para windows o WSL mejor

ssh copy-id [al usuario y eso] -> dar llave publica del usuario. Pedirá contraseña y no volverá a requerir pedirla nunca más. Crea archivo con llaves públicas de personas autorizadas.

Al reiniciar MV pierde las llaves. Borrar entrada para eso

VÍDEO - IMPORTANTE

SSH: Secure Shell. Con este servicio podemos autenticarnos, seguridad (datos van encriptados) y control de integridad de estos datos.

Surge como alternativa segura a telnet ya que telnet no encriptaba datos y cualquiera con acceso a la red, estaría viendo el tráfico.

Con ssh, tendremos un cliente que se conecta con un servidor (Arq cliente-servidor).

Imaginemos que un cliente quiere conectarse a un servidor. Para esto, cliente invoca al cliente ssh (un cliente) para conectarse a un servidor. Tras una negociacion y seguir el protocolo consigue conectarse al servidor. El servidor tiene sshd (un servicio) , un demon que está constantemente escuchando.

OJO con ssh nos podemos referir al cliente y al servicio, durante el vídeo, puede haber alguna ambigüedad entre ambos de no especificar nada.

Servidor podría ser cliente ssh de otro servidor (por ejemplo, un clúster de ordenadores).

Al configurar ssh, tendremos dos archivos que es bien importante diferenciar:

- /etc/ssh/ssh**d**_config -> Servicio
- /etc/ssh/ssh_config -> Cliente

@Zukii on Wuolah

INSTALACIÓN EN UBUNTU SERVER

Disciplina server hardening

Empezamos con Ubuntu Server

Comprobar que tenemos IP 192.168.56.105/24 (105 porque .15 diría que es un error) con
> ip addr

Usaremos apt para descargar ssh. Filtraremos por servidores

> apt search ssh | grep server

> sudo apt install openssh-server

Nos da error y para solucionarlo actualizamos con sudo apt update (a mi no me dió xd)

Volvemos a instalar openssh-server

Comprobar que el servicio se ha activado (el proceso) con:
> ps -Af | grep sshd
Si no inicia el servicio usamos > sudo /etc/init.d/ssh start

Para probarlo podemos conectarnos al servidor con nuestro cliente con
> ssh localhost
> yes (ahora localhost pasa a ser un host conocido)
> ISE

CTRL+D para salir

Podemos ver el directorio /home/ssh y vemos que tenemos un documento known_hosts en el que aparece el fingerprint de localhost (y los fingerprints de los hosts que conozcamos)

Es importante editar la configuración de ssh para impedir el acceso al root (ya que este podría acceder a todos los usuarios). Si quisiéramos acceder al root, tendríamos que usar un usuario como pasarela que cambiará de usuario y pasará a tener privilegios.

> sudo nano /etc/ssh/sshd_config

Aquí, buscamos #PermitRootLogin y ponemos no (por defecto prohibit-password, podemos acceder a root con cualquier mecanismo distinto a la contraseña)

> systemctl restart sshd (reiniciar servicio sshd)

Probamos a intentar acceder como root desde otra MV (por ejemplo WSL de Ubuntu) con
> ssh root@192.168.56.105 (aparece error de Permission denied)
> ssh -l root 192.168.56.105 (equivalente a la de arriba)

Si ponemos además de las órdenes anteriores, -v al final de la orden, podremos depurar problemas de conexión y ver que errores nos devuelve el servidor en caso de que haya problemas.

Ahora vamos a CentOS (tiene algunas diferencias)

ps -Af | grep sshd
La instalación por defecto de CentOS configura e inicia el servicio sshd como podemos ver

Diferencias	Ubuntu Server	CentOS
Instalación	Pregunta por la instalación	Por defecto
Comprobar estado del servicio (systemctl status sshd) NOTA: CTRL+D para salir	ssh (por eso usar solo segunda forma) y sshd	solo sshd
PermitRoot	no con contraseña	Sí
Firewall	ufw (uncomplicated firewall)	firewall-cmd
Firewall por defecto	No	Sí

Ahora probaremos a conectarnos al servicio desde otra terminal (WSL por ejemplo)
> ssh alvaro@192.168.56.110
> logout (para salir)

¿Podremos acceder al root del servicio de CentOS?
> ssh root@192.168.56.110 (y nos deja --> Diferencia)

>sudo nano /etc/ssh/ssh**d**_config

Buscamos el #PermitRootLogin y ponemos no.

> systemctl restart sshd (reiniciar servicio sshd)

Volvemos a probar:

> ssh root@192.168.56.110

Vemos como ya no nos deja. Ya tendremos configurado sshd en Ubutnu y CentOS e inhabilitado el acceso como root. Esto es lo mínimo que hay que hacer cuando estamos utilizando el servicio sshd

Ahora cambiaremos el puerto (por defecto ssh es el 22). Esto lo hariamos para evitar que sea trivial intentar acceder al servicio. Para ellos tenemos la directiva port y el archivo de configuración.

@Zukii on Wuolah

En este caso, en vez de usar nano o vi, usamos stream editor, nos permitirá buscar una cadena y sustituirla.

(Esto lo hacemos desde la WSL, siendo clientes del servicio de CentOS)

> ssh alvaro@192.168.56.110
> sudo su
> sed s/'Port22'/'Port 22022'/ -i /etc/ssh/sshd_config
(sustituir la cadena Port22 por Port22022 en el archivo sshd_config)
> systemctl restart sshd.

Ahora desde otra terminal (otra de WSL por ejemplo) y tratamos de acceder con
> ssh alvaro@192.168.56.110
Nos dejaría entrar igualmente por el puerto 22. (Nota, para elegir puerto en el comando ssh es con la opción -p [núm], en nuestro caso -p 22022)

El problema es que la cadena estaba comentada. Repetimos, pero quitando el #
(también podemos hacerlo con nano)
> sed s/'#Port22022'/'Port 22022'/ -i /etc/ssh/sshd_config
(sustituir la cadena Port22 po r Port22022 en el archivo sshd_config)
> systemctl restart sshd.

Nos da un error de Job for sshd…
Para monitorizar estos problemas tenemos el comando journalctl.
Comprobamos el estado de sshd:

> systemctl status sshd -> Vemos que hay error al iniciar el servicio. Al no tener más info invocamos a journalctl -xe

>journalctl -xe
Vemos que hay error al intentar asignar el puerto 22022 por permiso denegado. Editamos el archivo de configuración con nano y vemos que en la cabecera hay una línea que nos dice que si queremos cambiar el puerto en un sistema que ejecuta SELinux, hay que informar a SELinux sobre este cambio.

Lo haremos mediante el comando semanage (que permite gestionar políticas de SE Linux)

> semanage port -a (añadir puerto) -t ssh_port_t (tipo de puerto) -p tcp (protocolo) 22022

Instalaremos el paquete, buscaremos quien proporciona el comando semanage:
>dnf provides semanage

OJO, si nos da error de mirrorlist o algo así lo solucionamos con el comando dhclient

Vemos que el paquete policycoreutils-python… contiene el binario semanage

Podemos usar yum o dnf para instalar los paquetes (CentOS)
>dnf install policy…

@Zukii on Wuolah

> semanage port -l | grep ssh (Vemos los puertos y buscamos el de ssh)
> semanage port -a -t ssh_port_t -p tcp 22022
> semanage port -l | grep ssh (Vemos que se ha añadido el puerto de ssh al 22022)

Volvemos a intentar acceder (en WSL diferente a la que estamos como clientes: > ssh alvaro@192.168.56.110 -p 22022 -v) pero vemos que tenemos un error.

Salimos de WSL en la que estabamos como clientes y probamos a conectarnos desde el propio CentOS con:

>ssh localhost -p 22022

Y vemos que si nos deja. El problema es el firewall (Diferente entre Ubuntu y CentOS).
Configuraremos el firewall para que nos permita acceder al puerto 22022.

(En CentOS)

> sudo firewall-cmd --add-port 22022/tcp --permanent (añadimos puerto 22022 con protocolo tcp. Al añadir de forma permanente al recargar el firewall o reiniciar máquina, el puerto se abrirá)

> sudo firewall-cmd --reload (recargar firewall para que se abra ahora. Podíamos poner el comando de antes sin el --permanent para que se abriera en el momento)

Probamos de nuevo a conectarnos desde la WSL y vemos como sí podemos conectarnos
> ssh alvaro@192.168.56.110 -p 22022

VOLVEMOS A UBUNTU

Usamos ufw.

Modificamos archivo de configuración.

> sudo nano /etc/ssh/sshd_config
Descomentamos la Línea de Port y ponemos Port 22022

>systemctl restart sshd

Comprobar que tenemos acceso desde WSL indicando el puerto

> ssh alvaro@192.168.56.105 -p 22022

Pero esta vez nos deja. UFW está por defecto desactivado (Diferencia)
Activaremos el Firewall y añadiremos el puerto.

> sudo ufw status (está inactivo)
> sudo ufw enable

Volvemos a intentar conectarnos desde la WSL (otra pestaña nueva)
Y vemos que no nos deja.

@Zukii on Wuolah

Tendremos que permitir el tráfico en el puerto 22022
> sudo ufw allow 22022

Volvemos a intentar conectarnos desde la WSL y vemos como ya si nos deja.

FIN