**PRÁCTICA 2**

**Lección 1 (SSH y configuración de red)**

**Yum vs dnf (CentOS)**

Problemas yum (CentOS):

-Requiere mucha memoria para buscar paquetes y dependencias.

-Está escrito en Python -> es muy lento.

-Tiene módulos que no están completamente documentados.

Ventajas dnf:

-Módulos bien documentados.

-Mejora el rendimiento porque está escrito de forma híbrida con C++ (Lenguaje OO) y Python (Lenguaje interpretado).

**apt vs snap (Ubuntu)**

apt:

-Tiene un problema con las dependencias externas.

snap:

-Dependencias incluidas dentro del propio paquete.

-------------------------------------------------------------------------------------------------------------------------

\* *Mirar video P1 para poder hacer ping entre máquinas y host \**

**En clase lo hemos hecho de otra forma:**

-Instalamos ifconfig: sudo apt-get install net-tools

-Añadimos los adaptadores de red.

-ifconfig aparecen enp0s3 pero no enp0s8(nuevo adaptador). Sale en ip addr.

-modificamos netplan sudo vi /etc/netplan/00-installer-config.yaml

-modo insercion y añadimos enp0s8 (replicamos) y añadimos rango de direcciones 192.168.56.105/24(ubuntu) 192.168.56.110/24 (centOS)

-hay que aplicar el netplan -> sudo netplan apply

-comprobamos con ifconfig

-hacemos reboot

Entramos en CentOS:

-ifconfig, nos aparece enp0s8

-En centOS no hay netplan (es quisquilloso con tema redes) enp0s8 no tiene ninguna dirección ip asignada por lo que hay que asignarsela. Ahora la configuración de red está en /etc/sysconfig/network-scripts/ifcfg-enp0s8 (no está creado ese fichero)

BOOTPROTO=none

NAME=enp0s8

DEVICE=enp0s8

ONBOOT=yes //esto es lo que se tiene que poner yes para que se guarde la configuracion

IPADDR=192.168.56.110

NETMASK=255.255.255.0

reboot

-------------------------------------------------------------------------------------------------------------------------

Iniciamos UbuntuServer

Instalamos ssh-server: sudo apt-get install openssh-server

Hay que activar el sshd sudo systemctl start sshd y para desactivar sudo systemctl restart sshd

Comprobar que está bien instalado viendo el demonio de ssh: sudo systemctl status sshd

ssh-server viene instalado por defecto en CentOS

-------------------------------------------------------------------------------------------------------------------------

Podemos hacer ssh ahora desde nuestra terminal de host a las maquinas Ubuntu y CentOS

clear pero bien -> setterm -reset

-------------------------------------------------------------------------------------------------------------------------

Ahora añadimos seguridad

**Ubuntu**

sudo vi /etc/ssh/sshd\_config -> bajamos a #port 22 y lo descomentamos y ponemos a 22022 descomentar tambien permitrootlogin quitamos prohibit-password y ponemos no

¿Qué sentido tiene? Podemos hacer que el usuario acceda al sistema como **usuario** y después dentro del ssh hacemos sudo su y somos root. Hacer ssh root@192.... NO es seguro, lo suyo sería tener contraseñas distintas para root que para el usuario.

Comprobamos con sudo systemctl restart sshd, systemctl status sshd.

para entrar ahora en ubuntu por ssh -> ssh albertolg@192.168.56.105 -p 22022

**CentOS**

Igual que en ubuntu para modificar puerto y rootLogin

Al hacer restart da error por mapeo a puerto 22022 por falta de permisos. CentOS tiene capa de seguridad extra y no le gusta que cambien sus puertos. No deja hacerlo como root.

Usamos semanage (secure enhancer) Instalamos con sudo yum provides semanage y hacemos install del paquete que nos sale con python.

sudo semanage port -l | more

sudo semanage port -a -t ssh\_port\_t -p tcp 22022

sudo systemctl restart sshd

------------------------------------------------------------------------------------------------------------------------

Falta hacer que te pille la clave pública la máquina del usuario para no tener que introducir la contraseña y hacer que aunque alguien supiera la contraseña no puede entrar sin estar en tu máquina aun sabiendo la contraseña. (no lo hemos hecho)

**Lección 2 (Copia de seguridad y control de versiones)**

**Copias de seguridad**

Creamos un directorio prueba y creamos dentro tres ficheros

lsblk

*Lo que está entre paréntesis son comentarios*

***dd*** *-> hace backups*

**sudo dd if***(input file)***=/dev/sda2 of***(output file)***=sda2Backup.img** *(por poner una extensión)*

***cp*;**

**ls | cpio -ov > pruebaBackup.cpio** *-> comando que requiere algo de entrada***;**

**cpio -idv < pruebaBackup.cpio**

***tar* -> comprimir != empaquetar**

**tar cvz(***para comprimir, hay otras 5 formas de comprimir man tar* **)f nombre archivo o carpeta a comprimir.tar.gz**

**Para descomprimir -> tar -xvf ficheroADescomprimir.tar.gz**

¿Para qué comprimimos ficheros?

Velocidad a la hora de hacer los Backups.

**rsync -e(external) "ssh -p 22022" pruebaTAR.tar.gz albertolg@192.168.56.110:/home/albertolg (asi conserva el nombre del fichero que estamos mandando)**

**EXPLICADO EN PDF**

**Lección 3**

ssh keygen

configurar ssh github para subir cosas en examen

**Examen:** otro puerto, hacer ssh entre máquinas, medidas seguridad, subir capturas a repo de github.

3 opciones:

- Cambio de puerto

- Deshabilitar acceso a root

- Habilitar el intercambio de claves

- Desactivar el acceso por contraseña

- Restringir acceso a unos usuarios determinados

- Banear IPs sospechosas

Llevar dos máquinas (1 con interfaz de red configurada, otra sin interfaz de red configurada)

------------------------------------------------------------------------------------------------------------------------

**Objetivo: pasar ssh\_keygen (clave pública a privada) de una máquina a otra**

**UBUNTU**

ssh-keygen se genera la clave

si hacemos ahora ls .ssh si se genera la clave tanto privada como publica.

**Hacemos lo mismo en CENTOS**

**en centos**

Ahora hacemos ssh-copy-id (me permite copiar las claves que tenga actualmente en el directorio ssh y la manda a la máquina que queramos) albertolg@192.168.56.110 -p 22022

**en ubuntu**

**IGUAL QUE CENTOS PERO CON OTRO PUERTO**

**LS .SSH -> sale authorized keys**

**Lo que hemos hecho hasta ahora es habilitar el intercambio de claves.**

**DESACTIVAR ACCESO POR CONTRASEÑA**

**password authentification no tanto en centOS como ubuntu (recordar que hay que hacer systemctl restart sshd)**

Podemos probar todo esto creando un usuario nuevo en centOs/se hace igual en ubuntu, o lo que sea para ver que si pide contraseña

**RESTRINGIR EL ACCESO A UNOS USUARIOS DETERMINADOS**

me voy al sshd\_config y escribo: AllowUsers userquesea

**(solo podran acceder ciertos usuarios)**

**No la dejamos fija porque no nos interesa mucho solo para saber como hacerlo**

**sshfs** permite a mi máquina local mapear virtualmente un directorio de la máquina remota

ubuntu: sudo apt-get install sshfs

es como mount hacemos mkdir sshCentOS y hacemos sshfs albertolg@192.168.56.110:/home/albertolg sshCentOS/ -p 22022

entonces podemos tener una carpeta de centOS en Ubuntu pero en la terminal de ubuntu.

umount sshCentOS

**xforwarding** me permite que programas que sí tengan interfaz gráfica, poder ser ejecutados en el servidor que no tiene interfaz

es añadir -X al hacer ssh **(esta en video)**

**screen y otra cosa mas pero meh (esta en video)**

**FAILTOBAN (banear una IP tras un determinado número de fallos en la contraseña) en centOS**

sudo yum install epel-release

sudo yum install fail2ban

sudo systemctl status fail2ban

sudo systemctl enable fail2ban

sudo systemctl status fail2ban

sudo systemctl start fail2ban

los importantes son los jail

CD /ETC/FAIL2BAN

ls

sudo cp -a jail.conf jail.local

nos vamos a parte de servicios ssh

debajo de port ponemos: enabled = true

sudo systemctl restart fail2ban.service

ubuntu hacemos un ssh al usuario creado de antes y fallamos contraseña. nos vamos ahora a centOS

hacemos sudo fail2ban-client status sshd y te sale to la informacion del fallo.

hacemos que ubuntu falle 6 veces y baneao por hacker. **PERO COMO FAIL2BAN ES UN CACHONDO HA BANEAO EL USUARIO DEL PUERTO 22 NO EL DEL 22022.**

Ahora nos vamos al fichero que se ha hecho aqui que esta en el video porque ha ido follao \* y en port quitamos ssh y ponnermos 22022. salimos guardamos, reiniciamos , se lo traga y ya se suponne que el colega hacker esta baneao.

**ahora desbaneamos**

sudo failt2ban-client set sshd unbanip 192.168.56.105

y te sale 1 -> un unban que se ha hecho

compruebas el estado y te dice que desde que ha estado activo ha baneado a uno pero ahora po no tiene baneao a nadie

**y estamos desbaneaos en ubuntu**

**QUEDA INSTALACIÓN DE LAMP**

**CONFIGURAR SSH GITHUB EN MI MAQUINA HOST Y SABER HACERLO EN LA MÁQUINA VIRTUAL**

**COMO DEJAR LA MAQUINA COMO QUIERE EL PROFESOR:**

**PASSWORD AUTHETIFICATION = YES**