Apuntes Practica 2

Sesion 1 – Configurar Ubuntu y CentOS como servidor SSH

Sesion 2 – Acceso sin contraseña y seguridad

Preguntas de examen

- ¿Que significa SSH? Secure Shell
- Servidor utilizará el demonio **sshd**.
- ¿Cómo modificar un fichero sin abrirlo? Con el comando **sed (string editor)**
- ¿Viene **sshd** instalado por defecto en CentOS? SI.
- ¿Como es el la denominación que utiliza SELinux para definir el puerto SSH? **ssh_port_t**
- ¿Qué es y para que sirve SELinux? …
- ¿El firewall de CentOS viene el firewall activado por defecto? ¿Y en Ubuntu Server? En CentOS si, en Ubunto no.
- ¿Como se llama el firewall de Ubuntu Server? ¿Y en CentOS? **Uncomplicated Firewall** y **firewall-cmd** respectivamente.
- ¿Como podemos ver los puertos abiertos?
- ¿Que es **fail2ban**? Lee logs y actualiza iptables para prohibir ciertas Ips. Tiene un “server” demonito que realiza las tareas e interactuamos con él a través del “client” y de los archivos de configuración.
- ¿Que significa IP?
- ¿Que es rkhunter? Es como un antivirus, detecta ficheros maliciosos.
- ¿Para que sirven TMUX y SCREEN? Te permite abrir varios terminales en la misma venta y organizarlos a nuestro gusto. TMUX se pierde si se cierra el padre, y SCREEN no.
  - Screen -r -d → -r (reatach) -d (detach) → Recupera la sesion de screen y se asegura de cerrar dicha sesion donde estubiera abierta.
- IIS → Internet Information Service (Servidor por defecto de Windows Server)
- HTML → HyperText Markup Language
- XAMPP/LAMP → Linux/Windows Apache MySQL PHP
- ¿Puerto por defecto de MySQL? 3306
- ¿Puerto por defecto de mariadb? 3307

Ubuntu

1. Usamos el comando **tasksel** para instalar el paquete **openssh-server**. Tambien podemos usar **apt-get** para instalarlo.
2. Comprobamos que funciona el ssh entre el host anfitrion y la maquina virtual con Ubuntu server (previamente debe existir conexión entre ellos, probar con **ping**) realizando una conexión con el comando:
    - ssh [IP_EQUIPO_A_CONECTAR]

3. Configuración básica:
    i. Cambiamos el puerto
        - Accedemos al fichero **/etc/ssh/sshd_config** que es fichero para la configuración del servidor SSH, mientras que el fichero **/etc/ssh/ssh_config** es para cliente.
        - Usamos el comando **vi /etc/sshd_config** para modificarlo y en la linea donde pone **Port** cambiamos el puerto existente por el **22022**.
        - Realizamos de nuevo una conexión desde el anfitrión hacia la maquina virtual pero esta vez indicando el puerto, con el comando siguiente para comprobar que todo funciona perfectamente:
            - **ssh nombre@[IP_EQUIPO_A_CONECTAR]:20022**
            - **ssh [IP_EQUIPO_A_CONECTAR] -l nombre -p 22022**
    ii. Configurar cortafuegos uncomplicated-firewall que viene desactivado por defecto:
        - **ufw status** → Nos muestra el estado del uncomplicated-firewall.
        - **Ufw enable** → Activa el uncomplicated-firewall.
        - **Ufw disable** → Desactiva el uncomplicated-firewall.
        - **Ufw allow 22022** → Añadimos el puerto al uncomplicated-firewall.
    iii. Bloquear el root.
        - Modificamos la siguiente linea del fichero **/etc/shh/shhd_config** al valor de **no**:
            - **PermitRootLogin yes → PermitRootLogin no**
        - Relanzamos el servicio sshd
            - **systemctl restart sshd.service**
        - Comprobamos que la conexión desde el anfitrión mediante el usuario root no la permite.
            - **Ssh IP -p 22022 -l root** → Permission denied
    iv. Acceso sin contraseña
    v. Seguridad
        1. FAIL2BAN
            1. Instalamos fail2ban
                1. sudo apt-get install fail2ban
            2. Comprobamos que cuantas ips hay baneadas para el demonio sshd (ninguna inicialmente)
                1. fail2band-client status sshd
            3. Realizamos varias conexiones mediante ssh desde la otra maquina de forma erronea y volvemos a ejecutar el comando anterior. Veremos que se ha baneado (durante X tiempo) la ip debido a los intentos de conexión erroneos.
            4. Aun así nos dejara acceder de nuevo aunque estemos baneados debido a que hay que especificarle a jai2ban que puerto de escucha para el ssh es el 22022 y no el 22 que trae por defecto
                1. vi /etc/fail2ban/jail.conf
                    1. [sshd]
                        1. port ssh, 22022
            5. Si queremos desbanear una IP

1. fail2ban-client set sshd unbanip [IP_BANEADA]
2. RKHUNTER

vi.

CentOS

1. Comprobamos que funciona el ssh entre el host anfitrion y la maquina virtual con Ubuntu server (previamente debe existir conexión entre ellos, probar con **ping**) realizando una conexión con el comando:
    - ssh [IP_EQUIPO_A_CONECTAR]

2. Configuración básica:
    i. Cambiamos el puerto
        - Accedemos al fichero **/etc/ssh/sshd_config** que es fichero para la configuración del servidor SSH, mientras que el fichero **/etc/ssh/ssh_config** es para cliente.
        - Usamos el comando **sed** para modificarlo y en la linea donde pone **Port** cambiamos el puerto existente por el **22022**.
            - **sed s/'#Port 22'/'Port 22022'/ -i /etc/ssh/sshd_config**
        - Reiniciamos el servidio ssh con el comando:
            - **systemctl restart sshd.service**
        - Realizamos de nuevo una conexión desde el anfitrión hacia la maquina virtual pero esta vez indicando el puerto, con el comando siguiente para comprobar que todo funciona perfectamente:
            - **ssh nombre@[IP_EQUIPO_A_CONECTAR]:20022**
        - Solicitamos una ip con **dhclient**
        - Instalamos el paquete **yum install policycoreutils-python** para cambiar los puertos de escucha del SSH.
            - **Semanage port -l | grep ssh** → Muestra los puertos gestionados por ssh
            - **Semanage port -a -t ssh_port_t -p tcp 22022** → Añade el puerto 22022 como puerto de escucha del SSH.
                - **-a** → Añadir puerto
                - **-r** → Eliminar puerto
            - **Semanage port -l | grep ssh** → Comprobamos que se ha añadido correctamente el puerto 22022.
        - Ahora debemos de añadir el puerto 22022 al firewall que tiene CentOS instalado por defecto con el comando:
            - **Firewall-cmd –add-port=22022/tcp** → Incluye el puerto al firewall pero al reiniciar la maquina ya no estará ahí.
            - **Firewall-cmd --permanent-cmd –add-port=22022/tcp** → Añade el puerto de forma permanente al firewall, pero hay que reiniciar la maquina o hacer un reload para que funcione.
                - **Firewall-cmd –reload**
            -
    ii. Bloquear el root
    iii. Acceso sin contraseña
    iv. Seguridad
        1. Añadimos EPEL (Repositorio de distintos de paquetes que no vienen por defecto en CentOS)
            1. yum install epel-release
        2. Instalamos **fail2ban**
            1. yum install fail2ban
        3. Comprobamos si fail2ban esta activo
            1. systemctl status fail2ban
        4. Si no esta activo lo iniciamos

    1. systemctl start fail2ban
5. Comprobamos que cuantas ips hay baneadas para el demonio sshd (ninguna inicialmente)
    1. fail2band-client status sshd
6. Realizamos varias conexiones mediante ssh desde la otra maquina de forma erronea y volvemos a ejecutar el comando anterior. Veremos que se ha baneado (durante X tiempo) la ip debido a los intentos de conexión erroneos.
7. Aun así nos dejara acceder de nuevo aunque estemos baneados debido a que hay que especificarle a jai2ban que puerto de escucha para el ssh es el 22022 y no el 22 que trae por defecto
    1. cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
    2. vi /etc/fail2ban/jail.local
        1. Desconectamos las lineas del sshd y especificamos el puerto 22022
8. Si queremos desbanear una IP
    1. fail2ban-client set sshd unbanip [IP_BANEADA]

Sesion 3 – Crear servidor web Ubuntu Server y CentOS

UBUNTU

1. Ejecutamos **sudo taskshell**
2. Instalamos el paquete **lamp server**
3. Ponemos la contraseña de MySQL (**practicas,ISE**)
4. Comprobamos que todo esta funcionando, para ello podemos:
    1. systemctl status mysql
    2. mysql -u root -p → -p para que nos pida la password (**NO ESCRIBIR EN LA LINEA DE COMANDOS**)
    3. systemctl status php
    4. php -a → php en modo interactivo
    5. Abrimos un navegador desde otra maquina y ponemos como URL la ip del servidor apache.
    6. Curl localhost → Muestra el codigo <http> de la dirección pasada.

CENTOS

1. Instalamos yum install httpd
2. Comprobamos si todo funciona bien con:
    1. curl localhost
    2. systemctl status httpd
3. Abrimos un puerto
    1. firewall-cmd --permanent –add-port=80/tcp
    2. firewall-cmd --reload
4. Para que en el proximo reinicio el proceso aparezca como activo
    1. systemctl enable httpd
5. Instalamos ahora el MySQL cliente y servidor
    1. yum install mariadb
    2. yum install mariadb-server
    3. Lo activamos para el proximo reinicio
        1. systemctl enable mariadb
    4. Lanzamos el comando mysql_secure_installation

1. remove anonymous user
2. remove test database
3. access to id
4. reload privilege tables
5. mysql -u root -p

6. Instalamos ahora el PHP
    1. yum install php
        1. Comprobamos que funciona con
            1. php -a → Modo interactivo de PHP

Ahora vamos a comprobar que los tres componentes estan funcionando en conjunto correctamente. Para ello creamos el siguiente script.php

<index.php>

```
<?php
$enlace =   mysql_connect('localhost','usuario_mysql','contraseña_mysql');
if(!$enlace){
   die('No pudo conectarse: '.mysql_error());
}
  echo 'Conectado satisfactoriamente';
  mysql_close($enlace);
?>
```

Copiamos dicho fichero /var/www/html/index.php

Ahora si recargamos el navegador dará un error, debido a que hay que decirle a apache que index.php sea intepretado por el servidor y el resultado de su ejecución sea lo que mande al usuario. Debemos solucionarlo editando el fichero de configuración:

/var/httpd/conf/httpd.conf

En este fichero añadimos DirectoryIndex index.html, index.php

Ahora relanzamos el servicio con el comando:

systemctl restart httpd

Ahora debemos de instalar la biblioteca de conexión con MySQL

yum install php-mysql

Ahora añadimos la regla para permitir que el proceso httpd pueda acceder al puerto 3306

setsebool -P httpd_can_network_connect_db on

Y ahora tras 6 años de comandos si recargas el navegador funciona, cosa que en Ubuntu hubieramos tardado 5 minutos.