5. Seguridad y Alta disponibilidad

1. Permisos especiales

a. Abrir el entorno de Vagrant “vagrant_security” y conectarse mediante ssh a la VM.
b. Buscar los ficheros con permisos suid con: find / -perm -4000 -exec ls -ld {} \; 2> /dev/null
   ¿Qué tipo de ficheros/directorios son?
c. Buscar los ficheros on permisos sgid con: find / -perm -2000 -exec ls -ld {} \; 2> /dev/null
   ¿Qué tipo de ficheros/directorios son?
d. Buscar los ficheros con el sticky bit con el comando: find / -perm -1000 -exec ls -ld {} \; 2> /dev/null
   ¿Qué tipo de ficheros/directorios son?
e. Crear dos usuarios con el comando *adduser admin1* y *adduser admin2* aceptando todas las opciones por defecto
f. Crear un directorio en /home que será compartido por los dos usuarios admin con el comando mkdir administradores
g. Crear un grupo para los administradores ejecutando: *groupadd administradores*
h. Al directorio recién creado le cambiaremos el grupo a administradores ejecutando: *chgrp administradores administradores* y le daremos permisos especiales y de escritura al grupo: *chmod g+ws administradores/*
i. Añadiremos los dos usuarios al grupo administradores: *usermod -aG administradores admin1 y usermod -aG administradores admin2*. Verificamos con *getent group administradores*
j. Cambiaremos al usuario admin1 e intentamos crear un fichero en el directorio /home/administradores. ¿Qué permisos tiene?¿quien es el propietario y cual el grupo?
k. Cambiaremos al usuario admin2 y editamos el fichero creado por admin1. ¿puede modificarlo?
l. Repetiremos el ejercicio, pero en lugar de dar el permiso sgid le daremos el sticky bit al directorio administradores. Borraremos el directorio administradores con rm -r administradores y repetimos los pasos f al k cambiando el chmod g+ws por chmod o+wt. ¿que nos encontramos ahora?

2. Lynis - Hardening

a. Instalaremos la herramienta para verificar el bastionado de un servidor llamada lynis. Para ello desplegamos la VM del entorno vagrant_security y nos conectamos por ssh.

b. Una vez conectados nos descargamos el binario con wget desde la siguiente url: https://downloads.cisofy.com/lynis/lynis-2.7.5.tar.gz
c. Una vez descargado comprobamos el hash para verificar la integridad del fichero: sha256sum lynis-2.7.5.tar.gz el hash debe coincidir con 3d27ade73a5c1248925ad9c060024940ce5d2029f40aaa901f43314888fe324d
d. Descomprimimos el fichero *tar zxvf lynis-2.7.5.tar.gz*
e. Entrar al directorio y ejecutar la herramienta: cd lynis; ./lynis audit system
f. ¿que mejoras de seguridad se podrían hacer?

3. Instalación Keepalived

a. Arrancamos las dos máquinas del entorno vagrant_security e instalaremos keepalived y nginx en ambas máquinas ejecutando:
   sudo apt update
   sudo apt install -y keepalived nignx
b. Cambiaremos la página por defecto para poder hacer pruebas más adelante. Para ello ejecutamos **echo $(hostname) | sudo tee /var/www/html/index.nginx-debian.html** en las dos máquinas y el contenido de ese fichero contendrá el nombre de la máquina.
c. Verificamos que el nginx devuelve lo que esperamos, para ellos es suficiente con ejecutar **curl localhost** y nos devolverá el nombre de la máquina.
d. Pasamos a configurar el Keepalived. Si nos fijamos, las dos máquinas tienen una interfaz epn0s8 con una IP del rango 192.168.100.0/24. Vamos a configurar el keepalived para que la IP 192.168.100.200 esté activa en uno de los dos nodos. Para ello creamos el fichero **/etc/keepalived/keepalived.conf** con el siguiente contenido (cambiando MASTER o BACKUP en funcion de si es el nodo 1 o el nodo2):

```
vrrp_instance VI_1 {
        state MASTER o BACKUP
        interface enp0s8
        virtual_router_id 99
        priority 255
        advert_int 1
        authentication {
                auth_type PASS
                auth_pass keepcoding
        }
        virtual_ipaddress {
                192.168.100.200/24
        }
}
```

e. Guardamos y reiniciamos el servicio de keepalived con **sudo systemctl restart keepalived.service**.

f. Podemos verificar que ahora la IP 192.168.100.200 responde a ping y si hacemos curl nos devolverá el nombre de la máquina Master. Veremos que desde la máquina ubuntu1 en la interfaz enp0s8 tenemos las 2 IP levantadas: **ip -brief add show**
g. Vamos a probar el failover desde la máquina ubuntu2 vamos a dejar un ping lanzado a la IP de balanceo. Y desde la máquina ubuntu1 vamos a parar el servicio de keepalived. Si todo ha ido bien no deberíamos haber perdido ningún ping y la IP debería haberse movido al otro nodo.
h. Si ejecutamos de nuevo curl 192.168.100.200 vermos que nos devuelve el nombre de la segunda máquina.
i. Volvemos a arrancar el servicio en la máquina ubuntu1 (**sudo systemctl start keepalived.service**) y deberíamos volver a tener la IP configurada en esa: **curl 192.168.100.200**.

4. Instalación HAproxy

a. Seguimos con las mismas máquinas que en el ejercio anterior. En este caso, en una de ellas vamos a instalar el software HAproxy con **sudo apt install -y haproxy**.
b. Configuraremos el HAproxy editando el fichero /etc/haproxy/haproxy.cfg. Ya existe un fichero por defecto al que añadiremos las siguientes secciones antes de la secció defaults:

```
frontend keepcoding
        bind *:8080
        default_backend keepcoding_bend
        timeout client 50000ms


backend keepcoding_bend
        server ubuntu1 192.168.100.10:80 maxconn 32
        server ubuntu2 192.168.100.11:80 maxconn 32
        timeout connect 5000ms
        timeout server 50000ms
```

c. Verificamos que la configuración sea válida con **haproxy -c -f /etc/haproxy/haproxy.cfg** y en caso de no dar errores aplicaremos reiniciando el servicio **sudo systemctl restart haproxy.service**
d. Si ahora ejecutamos repetidamente **curl localhost:8080** veremos que nos está balanceando al nginx que hemos instalado anteriormente, cada vez a una máquina diferente.

5. Patroni