BORN2BEROOT

Link de la guía a seguir:
https://www.notion.so/Born2beRoot-119a7adf2b404b459fa565a3b0940912

PASOS

1. CONFIGURACIÓN DE LA MÁQUINA VIRTUAL

2. CONFIGURAR SSH EN VIRTUAL BOX

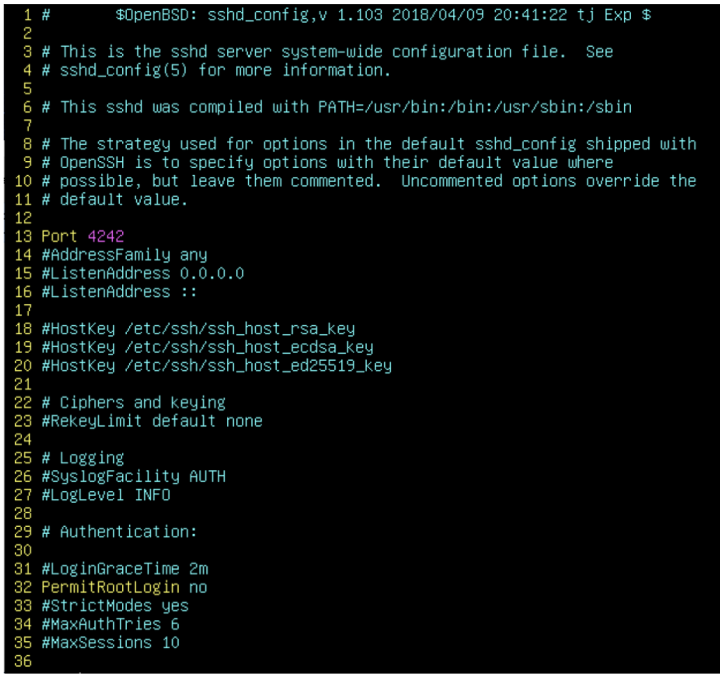
2.1. Configuración del protocolo SSH

Hay que ir a la raíz para editar eso, porque no tenemos derechos y con chmod no se pueden cambiar. → su. Para ir al usuario su zcanales.

Por defecto, SSH está configurado en el puerto 22, pero el proyecto nos pide que lo configuremos en el 4242. Para ello nos tenemos que ir a vi /etc/ssh/sshd_config

En este archivo cambiaremos dos cosas:

1. En la línea 13, cambiaremos el **puerto 22** por el **puerto 4242.** y quitamos comentarios
2. En la línea 32, cambiaremos los permisos de root para que no pueda entrar directamente desde SSH (viene especificado en el subject).



```
1  #       $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $
2
3  # This is the sshd server system-wide configuration file.  See
4  # sshd_config(5) for more information.
5
6  # This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
7
8  # The strategy used for options in the default sshd_config shipped with
9  # OpenSSH is to specify options with their default value where
10 # possible, but leave them commented.  Uncommented options override the
11 # default value.
12
13 Port 4242
14 #AddressFamily any
15 #ListenAddress 0.0.0.0
16 #ListenAddress ::
17
18 #HostKey /etc/ssh/ssh_host_rsa_key
19 #HostKey /etc/ssh/ssh_host_ecdsa_key
20 #HostKey /etc/ssh/ssh_host_ed25519_key
21
22 # Ciphers and keying
23 #RekeyLimit default none
24
25 # Logging
26 #SyslogFacility AUTH
27 #LogLevel INFO
28
29 # Authentication:
30
31 #LoginGraceTime 2m
32 PermitRootLogin no
33 #StrictModes yes
34 #MaxAuthTries 6
35 #MaxSessions 10
36
```

2.2. Configuración de UFW

Instalar sudo → apt install sudo (En root)

Instalar ufw → sudo apt install ufw (En root)

Ufw es Universal Firewall, un cortafuegos que básicamente sirve para abrir y cerrar puertos para que nuestra máquina pueda comunicarse por ellos. En esta página podemos ver cómo instalarlo y algunos comandos interesantes.

Una vez instalado, debemos comprobar si se encuentra activo o inactivo, para ello utilizamos el comando ufw status.

- Si está inactivo, lo iniciaremos con ufw enable y después reiniciaremos la máquina con reboot.
- Ahora debería encontrarse activo. Lo que vamos a hacer es cerrar el puerto 22 y abrir el 4242 con sudo ufw deny 22 y sudo ufw allow 4242

Para comprobar que todo está bien, volvemos a hacer un ufw status y debería salirnos esto

```
root@fportalo42:/etc/ssh# ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     DENY        Anywhere
4242                       ALLOW       Anywhere
22/tcp (v6)                DENY        Anywhere (v6)
4242 (v6)                  ALLOW       Anywhere (v6)
```

Vamos a limpiar un poco lo que hemos hecho. Para borrar puertos que no necesitamos lo haremos de la siguiente forma:

1. Utilizaremos sudo ufw status numbered para ver en qué posición tenemos los puertos. Aparecerá un listado numerado
2. Borraremos los puertos con sudo ufw delete NUM, siendo NUM el número del listado que nos da ufw status numbered. Nos pedirá que aceptemos

2.3. Comprobar que todo está correcto

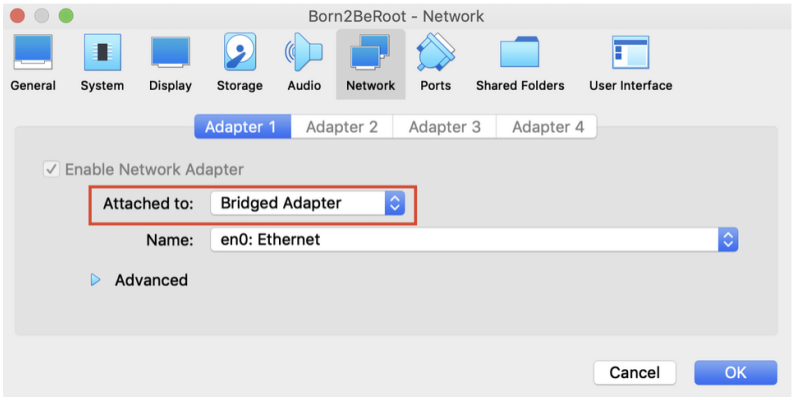
Una vez tenemos configurado nuestro archivo SSH y nuestro UFW, vamos a intentar conectarnos (fuera de virtualbox)

Para ello necesitamos saber la IP asignada de nuestra máquina. Dentro de nuestro servidor usamos el comando ip a. Saldrá parecido a esto. sudo ip a (desde root)

```
root@fportalo42:/etc/ssh# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:cd:06:62 brd ff:ff:ff:ff:ff:ff
    inet 10.11.200.144/16 brd 10.11.255.255 scope global dynamic enp0s3
       valid_lft 3633sec preferred_lft 3633sec
    inet6 fe80::a00:27ff:fecd:662/64 scope link
       valid_lft forever preferred_lft forever
root@fportalo42:/etc/ssh# _
```

El cuadrado rojo es la ip de la máquina, para conectarnos desde fuera, abriremos una terminal en nuestro ordenador local y escribir ssh fportalo@10.11.200.144 -p 4242 (sin /..) pero **antes hay que mirar la configuración de la network** en nuestra máquina virtual:

Tu máquina virtual tiene la red configurada en NAT. Para que funcione tu conexión, debes configurar el tipo de red desde VirtualBox. Para comprobarlo nos vamos a nuestra imagen en VirtualBox, pulsamos configuración, Network y cambiamos NAT dentro de *Attached to:* por Bridged Adapter: (hay que hacer sudo reboot para que se configure). Ir a Setting → Network



Los comandos son sencillos de entender. En fportalo pondremos nuestro usuario local, la ip del servidor y con el comando -p 4242 intentaremos acceder al puerto específico, en este caso el 4242. Debería salirnos lo siguiente.

ssh username@ip a -p 4242

```
c1r2s6% ssh fportalo@10.11.200.144 -p 4242
fportalo@10.11.200.144's password:
Linux fportalo42 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 27 13:24:27 2021 from 10.11.2.6
fportalo@fportalo42:~$
```

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: y
Please type 'yes', 'no' or the fingerprint: yes
```

👁️ ¡Ojo! El password utilizado es el de la máquina virtual

👁️ ¡Ojo! Es posible que en vez de salirte esto, se quede congelado el prompt. Significa que hay **algo** que **no está bien configurado.** Cosas que puedes probar son:

- Reinicia la máquina virtual con: reboot
- Comprueba que los puertos están bien configurados con ufw status. Utiliza ufw allow 4242 y ufw deny 22 para configurarlo y vuelve a reiniciar la máquina virtual con reboot

Vamos a instalar sudo y a configurar usuarios. MAL YA ESTÁ INSTALADO SUDO

SUDO Y ADMINISTRAR USUARIOS

Sudo (superuser do) es un usuario que tienen por defecto todos los sistemas operativos basados en Linux. Es un elemento imprescindible para la gestión y configuración de archivos sensibles, una capa más de seguridad. Este usuario es sensible porque puede hacer todos los cambios que quiera en el sistema operativo, y no todos deberían tener acceso a él.

El subject nos pide varias cosas:

- Instalar sudo
- Crear un grupo llamado **user42** y añadir nuestro usuario a los grupos **sudo** y **user42**
- Configurar la política de contraseñas para que sea segura

Instalar Sudo (Hecho en el anterior cap)

Vamos a instalar sudo, pues Debian solo viene por defecto con *su*, y el subject nos pide expresamente que instalemos *sudo.* Para ello, tan solo tenemos que utilizar el comando apt install sudo

Crear grupos

Para crear un grupo: groupadd user42 (No es zcanales42 → es user42 literal)

Para meter en un grupo a un usuario debemos respetar los antiguos grupos en los que se encontraba el usuario. Podemos comprobarlo con groups. Por tanto, para respetar los grupos que teníamos antes, debemos escribir el siguiente comando:

Para meter en un grupo : usermod fportalo -a -G user42

Una breve explicación de los comandos sería:

- groupadd : añade un nuevo grupo

- usermod_: incluye a un usuario en un grupo. Con el comando. Con el comando -a hacemos un append y con -G le decimos que lo haga a groups
- groups fportalo : sirve para comprobar los grupos en los que se encuentra el usuario en cuestión

Configurar Contraseña

Hay dos formas, a través de comandos y a través de archivos de configuración. Lo haremos desde un usuario con privilegios de sudo

Para cambiar el tiempo antes de que expire la contraseña: passwd -x 30 fportalo

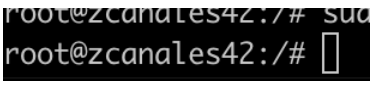
(-x, --maxdays MAX_DAYS → Set the maximum number of days a password remains valid. After MAX_DAYS, the password is required to be changed.)

Para cambiar el tiempo mínimo para cambiar la contraseña: passwd -n 2 fportalo.

(n, --mindays MIN_DAYS → Set the minimum number of days between password changes to MIN_DAYS. A value of zero for this field indicates that the user may change.)

Para cambiar el tiempo del mensaje de aviso de expiración: passwd -w 7 fportalo.

hay que ir a la carpeta raíz del usuario raíz →



Desde el archivo de configuración, tenemos que entrar en vim /etc/login.defs, ir bajando por el archivo hasta llegar a la parte de **Password aging controls**, y en la parte que no está comentada introducir la siguiente información.

```
#
# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   30
PASS_MIN_DAYS   2
PASS_WARN_AGE   7
```

Cambios en la política de contraseñas

Tenemos este artículo con documentación muy interesante. Vamos a comentarlo aquí:

1. Hemos de descargar el paquete con sudo apt -y install libpam-pwquality
2. Entramos en el archivo common-password con sudo vim /etc/pam.d/common-password
3. Añadimos las siguientes funciones especificadas en el subject. En el artículo salen explicados y en el archivo irían aquí:

   abrir con nano

```
# here are the per-package modules (the "Primary" block)
password        requisite                       pam_pwquality.so retry=3 minlen=10 maxrepeat=3 ucred
it=-1 dcredit=-1 difok=7 reject_username enforce_for_root
```

Hay otro archivo donde meter un elemento importante de configuración. Para entrar en él hemos de utilizar sudo vim /etc/sudoers y escribir otros defaults tal que así:

```
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:snap/bin"
Defaults        badpass_message="Bad boy, bad password"
Defaults        logfile="/var/log/sudo/sudo.log"
```

La información la he sacado de Sudoers Manual, dejo explicado aquí los comandos que hemos tocado:

- secure_path = hemos añadido al final del todo ":snap/bin". Esto se hace por seguridad
- badpass_mesaagge = sirve para enviar un mensaje de error cuando la contraseña no se ha introducido bien
- logfile = sirve para hacer un registro de veces que se utiliza sudo. Para que este comando funcione correctamente, hemos de crear la carpeta /var/log/sudo/ y dentro de esta, un archivo llamado sudo.log . Ahora, cada vez que nos demos permisos de sudo, se quedará registrado aquí. con mkdir y touch
- BONUS = Si escribes un default con insults, activas una funcionalidad en la cual cada vez que introduces mal la contraseña, tu sistema operativo te insulta 🤣

Una vez que terminamos la configuración de contraseñas, tenemos que cambiarlas todas, es decir, la del usuario **fportalo** y la del usuario **root.** ¡Esta pregunta tiene trampa! Si ya has cambiado las políticas de contraseña, puede que no te permita cambiar si no han pasado dos días desde que te hiciste la máquina virtual

Para comprobar las políticas de contraseña, lo que haremos será cambiar la fecha del sistema para que nos expire la contraseña, ver el mensaje de expiración a los siete días, y algunas otras funcionalidades que hemos implementado.

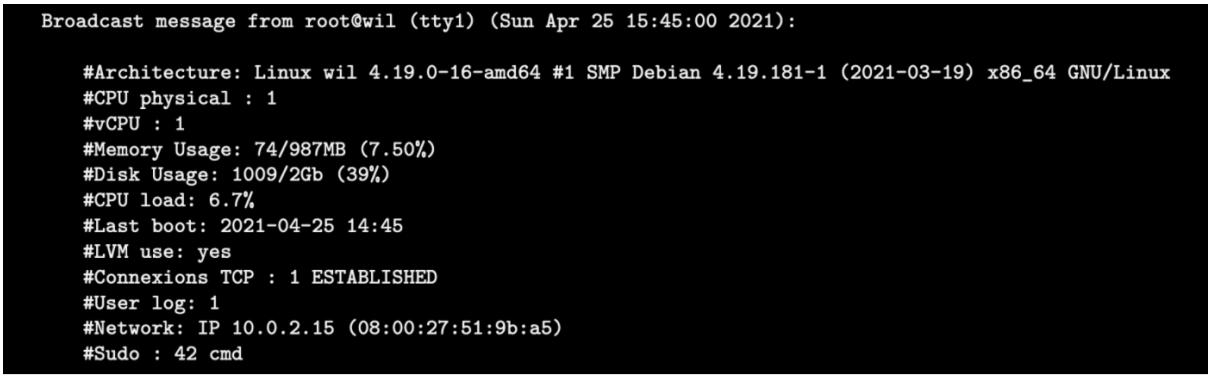
Cambio de fecha date 06152021 siendo 06 el mes, 15 el día, 20 la hora y 21 los minutos

Cambio de contraseña passwd

MONITORING.SH

Vamos a crear un script para mostrar algunas características y funciones de nuestra máquina virtual, como la Arquitectura, el último inicio de sesión o el número de comandos sudo utilizados . Para ello aprenderemos algunos comandos de terminal, como **awk**, **lscpu**, **lvdisplay**... Esta sección es muy interesante por la destreza que se adquiere navegando por la terminal, por eso nos pararemos en entender bien los comandos que utilizaremos.

Nuestro script debe mostrar la siguiente información:



Lo primero que tenemos que hacer es crear un archivo en la carpeta /root , esto es debido a que dentro de nuestro script tendremos que utilizar comandos que necesitan permisos de sudo y si lo hacemos en /home tendremos problemas a la hora de mostrar *LVM use* y *Sudo. P*or lo tanto, tendremos que acceder al sistema como **root***.* Ahora vamos a crearlo con touch /root/monitoring.sh y darle permisos con chmod.

```
#!/bin/bash



#Arquitecture

echo "#Architecture:" $(uname -a)

#CPU Physical

echo "#CPU physical:" $(lscpu | awk ' NR==5 {print $2}')

#Virtual CPU

echo "#vCPU :" $(lscpu | grep Socket\(s\) | awk '{print $2}')

#Memory Usage

free --mega | awk 'NR==2{printf "#Memory Usage: %s/%sMB (%.2f%%)\n", $3,$2,$3*100/$2
}'

#Disk Usage

df -h | awk '$NF=="/"{printf "#Disk Usage: %d/%dGB (%s)\n", $3,$2,$5}'

#CPU Load

top -bn1 | grep load | awk '{printf "#CPU Load: %.2f%s\n", $(NF-2), "%"}'

#Last Boot

echo "#Last boot:" $(who -b | awk ' {print $3,$4}')

#LVM
```

```
echo "#LVM use:" $(sudo lvm pvdisplay | grep Allocatable | awk '{print $2}')

#Connections TCP

echo "#Connetions TCP:" $(ss -s | grep TCP | awk 'NR==2 {printf "%d ESTABLISHED\n", $3}')

#User log

echo "#User log:" $(who | wc -l)

#Network IP

echo "#Network: IP" $(hostname -I) $(ip a | grep link/ether | awk '{printf " (%s)\n", $2}')

#Sudo

echo "#Sudo : " $(cat /var/log/sudo/sudo.log | grep USER | wc | awk '{printf "%s cmd\n", $1}')
```

Chmod

chmod es una herramienta que utilizamos para dar o quitar permisos de escritura, lectura y ejecución a los usuarios. La sintaxis es la siguiente:

Así, nosotros le pondremos a nuestro archivo ****monitoring.sh ****los permisos chmod 755 monitoring.sh.

- El **7** pertenece a los permisos del dueño
- El primer **5** a los permisos del grupo
- El segundo **5** a los permisos de otros

El script

Vamos a analizar parte por parte el script. Algunas partes tendremos que utilizar comandos en los cuales pararemos para poder usarlos en un futuro. Primero veremos cada apartado y al final del artículo tendremos el script entero completo

Arquitectura

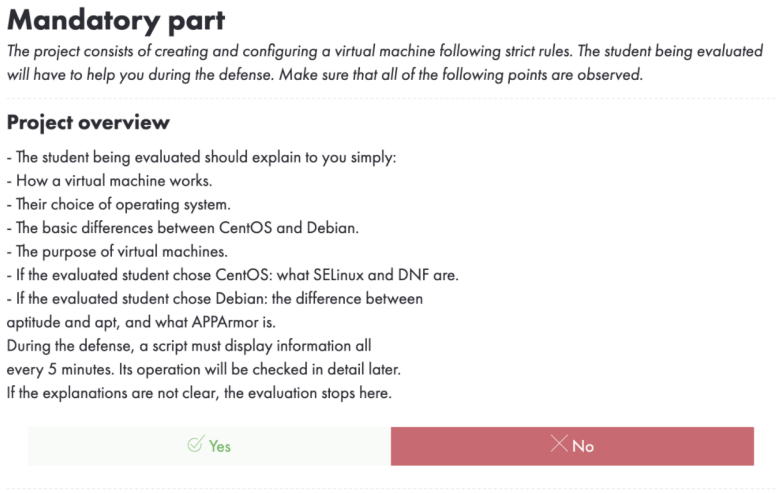
Para saber la arquitectura de nuestro sistema operativo basta con usar el comando uname -a, aunque antes tenemos que escribir "#Arquitecture: ", por tanto el comando quedaría así:

```
echo "#Arquitecture: " $(uname -a)
```

- $(uname -a) . Como ya sabemos, uname -a muestra la arquitectura, pero $( ) sirve para convertir una variable con aquello que muestre el comando que se encuentre dentro. Otro ejemplo podría ser $(echo "hola!") el cual mostraría únicamente hola!

Para las siguientes dos características, **CPU physical y vCPU**, utilizaremos dos comandos. lscpu sirve para ver cracterísticas de tu sistema operativo y de tu hardware. Si lo hacemos solo, nos mostrará por pantalla esto:

PREGUNTAS EXAM



Mandatory part

The project consists of creating and configuring a virtual machine following strict rules. The student being evaluated will have to help you during the defense. Make sure that all of the following points are observed.

Project overview

- The student being evaluated should explain to you simply:
- How a virtual machine works.
- Their choice of operating system.
- The basic differences between CentOS and Debian.
- The purpose of virtual machines.
- If the evaluated student chose CentOS: what SELinux and DNF are.
- If the evaluated student chose Debian: the difference between aptitude and apt, and what APPArmor is.

During the defense, a script must display information all every 5 minutes. Its operation will be checked in detail later.
If the explanations are not clear, the evaluation stops here.

Yes | No

¿Cómo funciona una máquina virtual?
Funciona como un emulador de consolas, su funcionamiento consiste en replicar un sistema operativo utilizando los recursos de nuestro ordenador, creando así otras.

Diferencias entre CentOS y Debian

- Centos → enfocado a Servidores
- Debian → enfocado a Usuarios

Centos es más actualizado pero más complejo (requiere conocimiento de servidores).

Propósito de una máquina virtual
Montar servidores, probar otros sistemas operativos, probar viruses, más seguridad…

Diferencias entre aptitude, apt y que es AppArmor
Debian utiliza **un administrador de paquete**s llamado “opkd”. Estos paquetes cuentan con otras **(dependencias)**los cuales s**on administrados con apt** (AdvantagePackageTool). Aptitude agrega nuevos comandos pero la principal **diferencia con apt es la interfaz.** (aptitude cer la lista de paquetes).

Apparmor es u**n gestor de privilegios** de usuarios que añade una caa de seguridad a tu sistema operativo.

Simple setup

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.
- Ensure that the machine does not have a graphical environment at launch.
A password will be requested before attempting to connect to this machine.
Finally, connect with a user with the help of the student being evaluated.
This user must not be root.
Pay attention to the password chosen, it must follow the rules imposed in the subject.
- Check that the UFW service is started with the help of the evaluator.
- Check that the SSH service is started with the help of the evaluator.
- Check that the chosen operating system is Debian or CentOS with the help of the evaluator.
If something does not work as expected or is not clearly explained,
the evaluation stops here.

Yes | No

Comprobar Sistema Operativo

- uname -a

Comprobar SSH status

- sudo systemctl status ssh

(NOtas) ¿Qué es SSH?

SSH es un **protocolo de seguridad** para conectarse **remotamente a un servidor** a través de una interface de texto a la que llamamos terminal, también es la forma más común de conectarse a un servidor Linux. La **conexión con el servidor** es un **modelo** cliente-servidor.

https://www.notion.so/SSH-Secure-Shell-a9e17bac9e324455ac43462bbfb515ad

UFW

- ufw allow 8080
- ufw status para comprobar el puerto 8080
- ufw status numbered para ver qué posiciones ocupan los puertos
- ufw delete NUM

User

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

The subject requests that a user with the login of the student being evaluated is present on the virtual machine. Check that it has been added and that it belongs to the "sudo" and "user42" groups.

Make sure the rules imposed in the subject concerning the password policy have been put in place by following the following steps.

First, create a new user. Assign it a password of your choice, respecting the subject rules. The student being evaluated must now explain to you how they were able to set up the rules requested in the subject on their virtual machine.
Normally there should be one or two modified files. If there is any problem, the evaluation stops here.

- Now that you have a new user, ask the student being evaluated to create a group named "evaluating" in front of you and assign it to this user. Finally, check that this user belongs to the "evaluating" group.

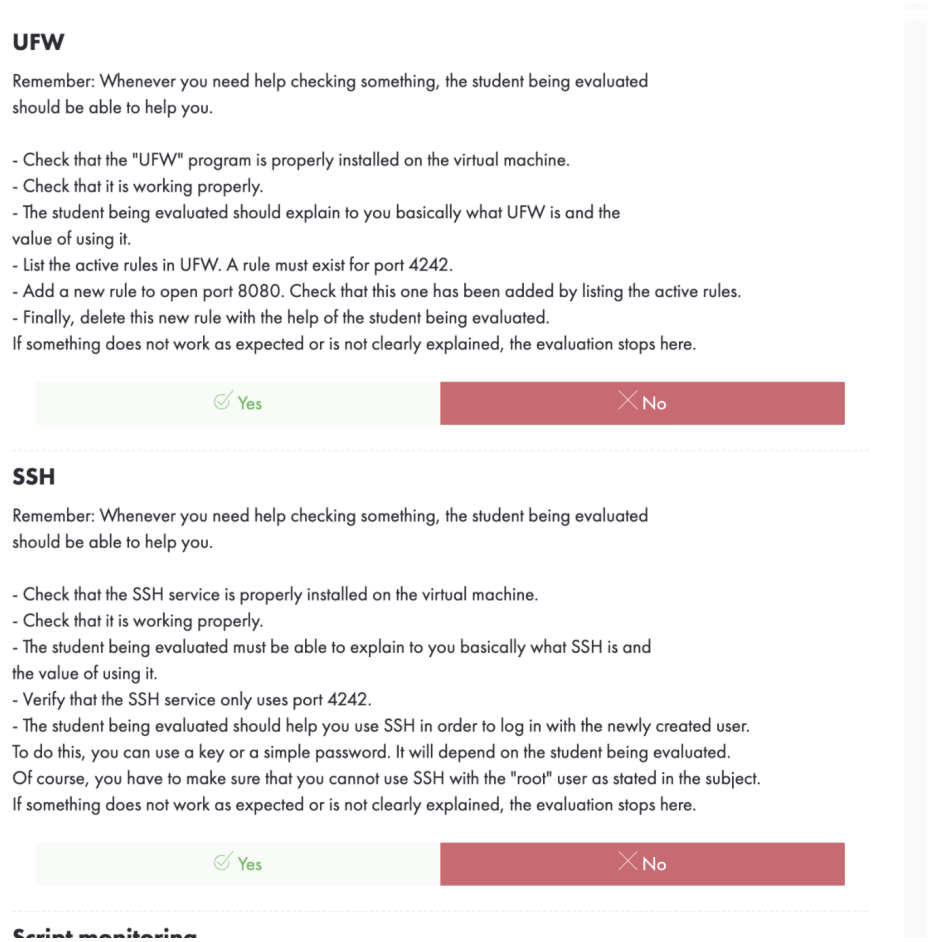
- Finally, ask the student being evaluated to explain the advantages of this password policy, as well as the advantages and disadvantages of its implementation. Of course, answering that it is because the subject asks for it does not count.

If something does not work as expected or is not clearly explained, the evaluation stops here.

Yes | No

Usuarios

- Añadir usuario : useradd seniorito_random añade un usuario nuevo. Tener cuidad con adduser , porque en algunos casos especiales pueden ser confusos con temas relacionados con la política de contraseñas, por ello es mejor usar useradd
- Añadir contraseña : passwd seniorito_random
- Añadir a un grupo : addgroup evaluating
- Añadir a un grupo : addgroup señor_random evaluating para añadir a una serie de grupos usermod seniorito_random -a -G evaluating



UFW

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

- Check that the "UFW" program is properly installed on the virtual machine.
- Check that it is working properly.
- The student being evaluated should explain to you basically what UFW is and the value of using it.
- List the active rules in UFW. A rule must exist for port 4242.
- Add a new rule to open port 8080. Check that this one has been added by listing the active rules.
- Finally, delete this new rule with the help of the student being evaluated.

If something does not work as expected or is not clearly explained, the evaluation stops here.

Yes | No

SSH

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

- Check that the SSH service is properly installed on the virtual machine.
- Check that it is working properly.
- The student being evaluated must be able to explain to you basically what SSH is and the value of using it.
- Verify that the SSH service only uses port 4242.
- The student being evaluated should help you use SSH in order to log in with the newly created user. To do this, you can use a key or a simple password. It will depend on the student being evaluated. Of course, you have to make sure that you cannot use SSH with the "root" user as stated in the subject.

If something does not work as expected or is not clearly explained, the evaluation stops here.

Yes | No

Script monitoring

UFW

- ufw allow 8080
- ufw status para comprobar el puerto 8080
- ufw status numbered para ver qué posiciones ocupan los puertos
- ufw delete NUM

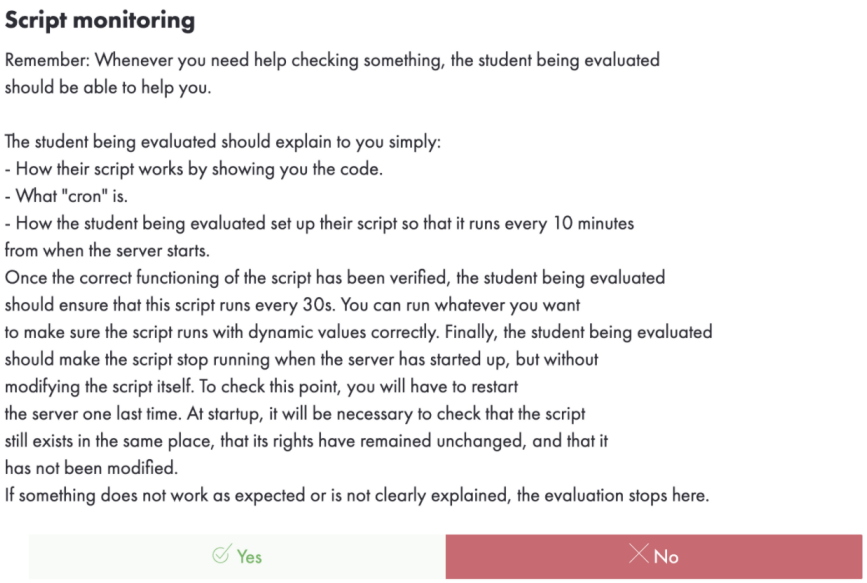
Comprobar SSH status

- sudo systemctl status ssh

  (NOtas) ¿Qué es SSH?

SSH es un **protocolo de seguridad** para **conectarse remotamente a un servidor** a través de una interface de texto a la que llamamos terminal, también es la forma más común de conectarse a un servidor Linux. La **conexión con el servidor** es un **modelo** cliente-servidor.

https://www.notion.so/SSH-Secure-Shell-a9e17bac9e324455ac43462bbfb515ad



cron

Las tareas cron de los sistemas Linux y Unix son **tareas programadas en el sistema** para que se **ejecuten cada tiempo determinado**

Crontab cada 30 segundos

https://crontab.guru/#10_*_*_*_*

Hay que hacer crontab -e y en el archivo hemos de escribir:

*/1 * * * * /home/fportalo/monitoring.sh | wall

*/1 * * * * sleep 30 ; /home/fportalo/monitoring.sh | wall

El comando **wall sirve para enviar por todas las terminales el input que le enviemos,** en este caso, el texto generado por monitoring.sh

TEORÍA

SSH **es un protocolo de seguridad** para **conectarse remotamente** a **un servidor** a través de una **interface de texto** a la que llamamos **terminal,** también es la forma más común de conectarse a un servidor Linux.

La conexión con el servidor es un **modelo cliente-servidor.** Esto significa que el lugar desde donde enviamos la información a la máquina en remoto es el **cliente** y la máquina en remoto que ejecuta los comandos es el **servidor**.

El Servidor

Este servidor está ejecutando un software llamado **SSH daemon.** Un daemon es simplemente un proceso ejecutado en segundo plano**.** Este software escucha las conexiones en un puerto específico y autentifica las peticiones de conexión o las rechaza si el usuario tiene las acreditacciones correctas

El Cliente

El ordenador del usuario que se quiere conectar ha de tener un cliente SSH, un software que sabe cómo comunicarse con el servidor mandándole información para conectarse con él como el usuario y las credenciales

Cortafuegos / Firewall

https://kinsta.com/es/blog/que-es-un-cortafuegos/

- Es una barrera entre un ordenador y el «mundo exterior».
- ¿Cómo funciona exactamente un cortafuegos?

  Los cortafuegos simplemente monitorean el tráfico entrante y saliente de un dispositivo, escaneando en busca de cualquier señal de actividad maliciosa. Si detecta algo sospechoso, lo bloqueará instantáneamente para que no llegue a su destino.

  Es un gran sistema de filtración para tu ordenador o servidor.

- El cortafuegos puede protegerte a ti o a tu sitio web frente a:

- Intrusiones: Los cortafuegos impiden a los usuarios no autorizados acceder a tu ordenador o servidor de forma remota y hacer lo que quieran.
- Malware: Los atacantes que logran infiltrarse pueden enviar malware para infectarte a ti o a tu servidor. El malware puede robar información personal, propagarse a otros usuarios o dañar de alguna manera tu computadora.
- Ataques de fuerza bruta: Intentos de los hackers de probar cientos de combinaciones de nombres de usuario y contraseñas para descubrir las credenciales de acceso de su administrador (o de otros usuarios).
- Ataques DDoS: Los cortafuegos (especialmente los de aplicaciones web) pueden intentar detectar la afluencia de tráfico falso que se produce durante un ataque DDoS.

IP

https://www.xataka.com/basics/que-es-una-direccion-ip-y-como-puedes-saber-la-tuya

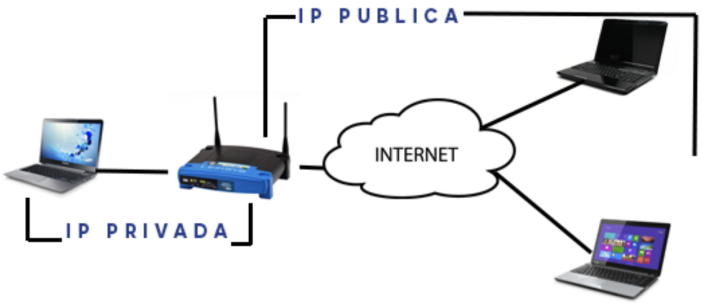
Se tratan e una especie de "**matrícula**" para identificarte cuando estás conectado. Sin embargo hay dos tipos de direcciones IP, las IP Públicas y las IP Privadas, y cada una de ellas tiene una finalidad totalmente diferente.

- IP Pública es la dirección que te asigna tu ISP, empresas que dan acceso a Internet como Telefónica, Vodafone, etcétera, y sirve para identificarte dentro de Internet cuando te conectas. Aunque también las hay fijas, es común que estas IPs suelan ser dinámicas y vayan cambiando sin que te des cuenta cada cierto tiempo.

Nadie puede navegar por la red sin una IP, y ninguna página web puede estar online si no tiene una IP asociada. De hecho, cuando tú escribes una dirección como 'www.google.es', lo que hace el navegador es traducir ese texto a una dirección IP para poder conectarse a la página de Google y acceder a su contenido.

Por lo tanto, estas direcciones IP públicas son como la matrícula que se te asigna cuando te conectas. Es una manera de identificarte como usuario en la inmensidad de la red, ya que ninguna IP se puede repetir.I

- IP Privadas, que son las que se utilizan en redes privadas como la que creas en tu casa conectando varios dispositivos a través de tu WiFi. Cuando lo haces, cada dispositivo como tu impresora, tu router o tu smartphone tiene una IP propia, y para que no haya conflictos ellos cada uno tendrá una IP diferente



SUDO

https://www.fayerwayer.com/2019/10/como-usa-el-programa-sudo/

El programa Sudo (super user do, en Inglés) es una utilidad de los sistemas operativos tipo Unix, como Linux, BSD, o Mac OS X, que permite a los usuarios ejecutar programas con los privilegios de seguridad de otro usuario (normalmente el usuario root) de manera segura, convirtiéndose así temporalmente en súper usuario.

Si un usuario normal desea ejecutar un comando de root (o de cualquier otro usuario), Sudo verifica en su lista de permisos y si está permitido la ejecución de ese comando para ese usuario, entonces Sudo se encarga de ejecutarlo", explica Linuxtotal.