Defensa de Born to be root

General Instruction

Para este apartado necesitamos extraer la número que debe coincidir con **signature.txt** que es el único archivo en el repositorio del proyecto.

Para extraer este número debemos entrar en la siguiente ruta: **/sgoinfre/goinfre/Perso/alruiz-c/born2beroot/Borntoberoot** y ejecutamos **shasum Borntoberoot.vdi**. Este paso tardará unos 10 minutos.

Mandatory Part

¿Cómo trabaja una máquina virtual?

En muchos sentidos, una máquina virtual (VM para abreviar) es prácticamente idéntica a un equipo físico normal. Ambos tienen una CPU, algo de RAM, un disco duro y una conexión a Internet si es necesario, y se pueden instalar varios sistemas operativos y software en ellos. La única diferencia es que un equipo física usa su propio hardware, mientras que una máquina virtual toma prestados los recursos físicos de su anfitrión. Por lo tanto, una máquina virtual es solo código, una computadora virtual dentro de un servidor físico. El software llamado hipervisor o monitor de máquina virtual (VMM) es responsable de crear y ejecutar máquinas virtuales, así como de administrar recursos, como CPU, memoria y almacenamiento, entre ellos.

El proyecto Born2beroot del Cursus de 42 explora los fundamentos de la administración de sistemas invitándonos a instalar una máquina virtual con VirtualBox. El servidor que vamos a crear debe tener el esquema de partición definido en el enunciado del ejercicio así como el sistema operativo Linux de nuestra elección: CentOS o Debian.

Nuestra opción de Sistema Operativo

Mi elección ha sido Debian.

Diferencias básicas entre CentOS y Debian

- Debian es más fácil de instalar que CentOS.

- Debian no tiene versión Enterprise.
- CentOS tiene menos actualizaciones que Debian pero las de Debian son más frecuentes.

El objetivo de una máquina virtual

Poder probar diferentes sistemas operativos asegurándonos que podemos hacer cualquier cosa sin dañar realmente ningún hardware de nuestro equipo. También podemos tener la VM en un pendrive, por ejemplo.

Diferencia entre APT y Aptitude

- APT no tiene interface gráfica, Aptitude sí.
- APT viene instalada de base en las instalaciones de Linux. En este proyecto siempre uso APT.

APPArmor

Este programa es un programa de seguridad. Para comprobar que está en uso ejecutamos: **sudo aa-status**.

Recordar que debe saltar un script cada 10 minutos

Simple Setup

- Si ya estoy logueado hacer **exit**. Volver a loguearme con un usuario que no sea root y usando una contraseña que se adecue a los que especifica el enunciado (10 caracteres, 1 mayúscula y 1 número)
- Que UFW esté iniciado ejecutando: **sudo ufw status**.
- Que SSH esté iniciado ejecutando: **sudo systemctl status ssh**.
- Asegurarse que usamos Debian ejecutando: **uname -a**.

User

- [ ] Crear grupo user42 y añadir mi usuario. sudo groupadd user42. sudo gpasswd -a alruiz-c user42

- Asegurarse que el usuario pertenece a los grupos “sudo” y “user42” ejecutando: **groups alruiz-c**.
- Mostrar cómo cambiar la política de contraseñas ejecutando: **sudo nano /etc/security/pwquality.conf**.
    - difok = 7 // Número de caracteres en la nueva contraseña que no deben aparecer en la antigua contraseña.
    - minlen = 10 // Mínimo de caracteres.
    - dcredit = -1 // Al menos un dígito.
    - ucredit = -1 // Al menos una mayúscula.
    - maxrepeat = 3 // Máximo de caracteres consecutivos repetidos en la nueva contraseña.
    - Userchek = 1 // La contraseña no puede contener el nombre de usuario.
    - retry = 3 // Reintentos para devolver error en caso de fallar contraseña.
    - enforce_for_root // Habilitado para usuario root.
- También mostrar este archivo en relación a política de contraseñas: **sudo nano /etc/login.defs**
    - PASS_MAX_DAYS 30 // Número de días máximo de uso de la contraseña.
    - PASS_MIN_DAYS 2 // Número de días mínimo de uso de la contraseña.
    - PASS_WARN_AGE 7 // Te avisa 7 días antes de que la contraseña expire.
- Crear un usuario y asignarle una contraseña con los siguientes comandos: **sudo useradd prueba**. Cambiamos contraseña con: **sudo passwd prueba**.
- Crear un nuevo grupo llamado evaluating con el comando: **sudo groupadd evaluating**. añadir el usuario prueba al grupo evaluating: **sudo gpasswd -a prueba evaluating**. Comprobar que el usuario está en el grupo: **groups prueba**.

- Explicar la ventaja de la política de contraseñas usada, en definitiva para crear una contraseña fuerte.

Hostname y partitions

- Comprobar que el hostname sea alruiz-c42: **hotsnamectl status**.
- Modificar hostname y reiniciar la máquina para verificar que se ha actualizado: **sudo hostnamectl set-hotname alruiz-c43**. Reiniciamos la máquina y hacemos un **hostnamectl status**.
- Restaurar hostname: **sudo hostnamectl set-hotname alruiz-c42**.
- Ver particiones de la VM ejecutando: **lsblk** y compararla con el enunciado.

Sudo

- Comprobar que Sudo está correctamente instalado en la VM ejecutando: **sudo —version**. Es obvio que está instalado correctamente ya que lo llevamos usando durante toda la evaluación.
- Agregar el usuario prueba al grupo sudo: **sudo gpasswd -a prueba sudo**. Comprobar con: **groups prueba**.
- Explicar que 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. Esto llevamos haciéndolo durante toda la evaluación con todos los comando que se preceden con la palabra sudo.
- Comprobar que existe la ruta /var/log/sudo. ejecutamos: **sudo nano sudo.log** cerramos y volvemos a ejecutar y vamos a la última linea para ver cual ha sido el último log con sudo, habrá 2 seguidos

UFW

- Comprobar que UFW está instalada ejecutando: **sudo ufw status**. Aquí vemos que está habilitado el puerto 4242.

- Explicar que UFW es un firewall que monitorea el tráfico de datos entre un equipo local y una red. Decide si permite o bloquea el tráfico.
- Crear una nueva regla para abrir el puerto 8080: **sudo ufw allow 8080**. Y comprobarlo: **sudo ufw status**.
- Eliminar esta regla: **sudo ufw delete allow 8080**. Y comprobarlo: **sudo ufw status**.

SSH

- Comprobar que SSH está instalado: **sudo systemctl status ssh**. Aquí podremos ver que está activo y que usa el puerto 4242.
- Conectar mediante SSH con el usuario prueba. Abrimos nueva terminal y ejecutamos: **ssh prueba@localhost -p 4242**.

Scrpit monitoring

- Comprobar el archivo monitoring.sh en la ruta /usr/local/bin/.
- Explicar que Crontab es un programa que ejecuta scripts de forma automática cuando lo indiquemos. Ejecutando: **sudo crontab -e**. Podemos observar que se ejecuta cada 10 minutos y la ruta del script a ejecutar.
- También explicamos el script llamado sleep.sh que nos sirve para calcular 10 minutos exactamente desde que se arranca la VM.

Bonus

- Comprobar con **lsblk** que las particiones son acordes al bonus.
- Comprobar que se ha instalado Wordpress. Miramos en el navegador **122.0.0.1:8080** y vemos que funciona wordpress.
- Comprobar que se ha instalado fail2ban ejecutando: **sudo systemctl status fail2ban**. Y probamos con los siguientes comandos el funcionamiento:

  $ sudo fail2ban-client status
  $ sudo fail2ban-client status sshd
  $ sudo tail -f /var/log/fail2ban.log

Intentamos conectar mediante ssh en una ventana de la consola ejecutando: **ssh prueba@localhost -p 4242** pero metemos mal la contraseña para ver cómo crea logs al fallar.