Tarea 1: Seguridad en arquitecturas Linux

1. Proponer una cuestión para cada una de las 3 etapas: Temprana, Intermedia y final.

Etapa temprana: ¿Puedo conseguir una lista de correos de la organización?

Etapa intermedia: ¿Los usuarios están bien formados? ¿Puedo conseguir acercarme a mis objetivos por medio de phising?

Etapa final: ¿Puedo tomar el control del sistema?

2. Elegir las tres cuestiones que consideres más importantes para cada etapa. Justificar la respuesta indicando algún ataque.

En la etapa temprana no me quedo con ninguna en concreto, ya que creo que toda información que se pueda obtener de manera “sencilla” siempre será una ventaja para el atacante. Para limitar vectores de amenaza es recomendable hacer informes internos con herramientas OSINT para comprobar que información es visible desde la red y en caso de detectar alguna amenaza, tratar de corregirla.

En la etapa intermedia, me quedo con la pregunta ¿Software con vulnerabilidades conocidas? Ya que, si en nuestras organizaciones no tenemos en cuenta esto, estamos dejando una puerta abierta muy clara y es muy probable que tengamos otros muchos problemas de ciberseguridad. Controlar lo conocido es muy importante.

Y en la etapa final me quedo, con ¿Cómo puedo explotar los recursos? Si a esta pregunta la dejamos sin respuesta, aunque hayan llegado hasta esta etapa es probable que tengamos tiempo para detectarlo, corregir la visibilidad de nuestros recursos y evitar un ataque.

Seminario 1:

1. Seguir los pasos del seminario anterior sobre la máquina virtual base. Entregar la salida por consola de lanzar los siguientes comandos en la máquina.

• find /opt -printf "%f: %p: %u: %g %m (%M) \n"

```
root@ubuntu:/# find /opt -printf "%f: %p: %u: %g %m (%M) \n"
opt: /opt: root: root 755 (drwxr-xr-x)
systemA: /opt/systemA: root: root 755 (drwxr-xr-x)
registroVentas.csv: /opt/systemA/registroVentas.csv: root: root 644 (-rw-r--r--)
addVenta.sh: /opt/systemA/addVenta.sh: root: root 755 (-rwxr-xr-x)
registroCompras.csv: /opt/systemA/registroCompras.csv: root: root 644 (-rw-r--r--)
addCompra.sh: /opt/systemA/addCompra.sh: root: root 755 (-rwxr-xr-x)
addConsulta.sh: /opt/systemA/addConsulta.sh: root: root 755 (-rwxr-xr-x)
registroConsultas.csv: /opt/systemA/registroConsultas.csv: root: root 644 (-rw-r--r--)
storm: /opt/storm: storm: componentes 700 (drwx------)
conf: /opt/storm/conf: storm: componentes 700 (drwx------)
storm.conf: /opt/storm/conf/storm.conf: storm: componentes 600 (-rw-------)
bin: /opt/storm/bin: storm: componentes 700 (drwx------)
storm.bin: /opt/storm/bin/storm.bin: storm: componentes 700 (-rwx------)
tmp: /opt/storm/tmp: storm: componentes 700 (drwx------)
db: /opt/db: db: db 700 (drwx------)
conf: /opt/db/conf: db: db 700 (drwx------)
db.conf: /opt/db/conf/db.conf: db: db 600 (-rw-------)
bin: /opt/db/bin: db: db 700 (drwx------)
db.bin: /opt/db/bin/db.bin: db: db 700 (-rwx------)
tmp: /opt/db/tmp: db: db 700 (drwx------)
tarballs: /opt/tarballs: root: root 755 (drwxr-xr-x)
db.tgz: /opt/tarballs/db.tgz: root: root 644 (-rw-r--r--)
web.tgz: /opt/tarballs/web.tgz: root: root 644 (-rw-r--r--)
storm.tgz: /opt/tarballs/storm.tgz: root: root 644 (-rw-r--r--)
web: /opt/web: web: componentes 750 (drwxr-x---)
conf: /opt/web/conf: web: componentes 750 (drwxr-x---)
web.conf: /opt/web/conf/web.conf: web: componentes 640 (-rw-r-----)
bin: /opt/web/bin: web: componentes 750 (drwxr-x---)
web.bin: /opt/web/bin/web.bin: web: componentes 740 (-rwxr-----)
tmp: /opt/web/tmp: web: componentes 750 (drwxr-x---)
```

• cat /etc/passwd

```
root@ubuntu:/# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
ubuntu:x:1000:1000:Ubuntu,,,:/home/ubuntu:/bin/bash
dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:109:65534::/var/run/sshd:/usr/sbin/nologin
web:x:1001:1001::/opt/web:/usr/sbin/nologin
storm:x:1002:1002::/opt/storm:/usr/sbin/nologin
db:x:1003:1003::/opt/db:/usr/sbin/nologin
```

2. Elaborar una lista de los principales comandos utilizados junto con una breve descripción de qué hace cada uno.

Creación de usuarios con el comando useradd:

```
root@ubuntu:/home/ubuntu# useradd web
root@ubuntu:/home/ubuntu# useradd storm
root@ubuntu:/home/ubuntu# useradd db
```

Creación de grupos con el comando groupadd y añadirles usuarios con usermod:

```
root@ubuntu:/home/ubuntu# groupadd componentes
root@ubuntu:/home/ubuntu# usermod -aG componentes web
root@ubuntu:/home/ubuntu# usermod -aG componentes storm
```

Cambiar la Shell de los usuarios con chsh:

```
root@ubuntu:/home/ubuntu# chsh web -s /usr/sbin/nologin
root@ubuntu:/home/ubuntu# chsh storm -s /usr/sbin/nologin
root@ubuntu:/home/ubuntu# chsh db -s /usr/sbin/nologin
```

Asignar la carpeta home de cada usuario:

```
root@ubuntu:/home/ubuntu# usermod -d /opt/web web
root@ubuntu:/home/ubuntu# usermod -d /opt/storm storm
root@ubuntu:/home/ubuntu# usermod -d /opt/db db
```

Estado de los usuarios:

```
root@ubuntu:/home/ubuntu# cat /etc/passwd | grep web:
web:x:1001:1001::/opt/web:/usr/sbin/nologin
root@ubuntu:/home/ubuntu# cat /etc/passwd | grep storm:
storm:x:1002:1002::/opt/storm:/usr/sbin/nologin
root@ubuntu:/home/ubuntu# cat /etc/passwd | grep db:
db:x:1003:1003::/opt/db:/usr/sbin/nologin
```

Tras descomprimir los tarballs con el comando tar vamos a cambiar los permisos de carpetas y ficheros:

```
root@ubuntu:/opt/web# find . -type d -exec chmod 750 {} +
root@ubuntu:/opt/web# find . -type f -exec chmod 640 {} +
root@ubuntu:/opt/web# find ./bin -type f -exec chmod 740 {} +
```

Ver estado de servicios:

```
root@ubuntu:/etc/systemd/system# ps -auwx | grep web
web       1428  0.0  0.3  12520  3096 ?        Ss   10:37   0:00 /bin/bash /opt/web/bin/web.bin
web       1820  0.0  0.0   7292   752 ?        S    10:38   0:00 sleep 1
root      1826  0.0  0.0  14224   924 tty1     S+   10:38   0:00 grep --color=auto web
```

3. De la lista de cuestiones que se pueden encontrar en la sección 1.2, seleccionar aquellas que están relacionadas con una mala gestión del sistema de permisos.

ETAPA TEMPRANA

- ¿usa recursos externos?, ¿provee recursos?

ETAPA INTERMEDIA

- ¿qué privilegios consigo?

- ¿qué recursos internos hay en la máquina?

- ¿necesito escalar privilegios?

- ¿backdoor back-up?

ETAPA FINAL

-¿cómo puedo explotar los recursos?

- ¿necesito extraer recursos?

- ¿cómo puedo extraerlos?

- ¿puedo destruir recursos?

Seminario 2:

1. Seguir los pasos del seminario anterior sobre la máquina virtual base. Entregar la salida del siguiente comando ejecutado en la máquina virtual.

• getfacl /opt/systemA

```
root@ubuntu:/opt/systemA# getfacl /opt/systemA
getfacl: Removing leading '/' from absolute path names
# file: opt/systemA
# owner: root
# group: root
user::---
group::---
group:administradores:rwx
group:consultores:r-x
group:operadores:r-x
mask::rwx
other::---
```

2. Elaborar una lista de los principales comandos utilizados junto con una breve descripción de que hace cada uno.

Iniciamos el ejercicio quitando todos los permisos al directorio de trabajo:

```
root@ubuntu:/home/ubuntu# chmod -R 000 /opt/systemA
root@ubuntu:/home/ubuntu# ls -la /opt/systemA
total 32
d--------- 2 root root 4096 Dec  7  2017 .
drwxr-xr-x 7 root root 4096 Jun 15  2021 ..
---------- 1 root root   83 Dec  7  2017 addCompra.sh
---------- 1 root root   89 Dec  7  2017 addConsulta.sh
---------- 1 root root   80 Dec  7  2017 addVenta.sh
---------- 1 root root   24 Dec  7  2017 registroCompras.csv
---------- 1 root root   30 Dec  7  2017 registroConsultas.csv
---------- 1 root root   21 Dec  7  2017 registroVentas.csv
```

Con el comando setfacl creamos la lista de acceso:

```
root@ubuntu:/opt/systemA# setfacl -Rm g:administradores:rwx /opt/systemA_
```

```
root@ubuntu:/opt/systemA# setfacl -m g:operadores:rx /opt/systemA
root@ubuntu:/opt/systemA# setfacl -m g:consultores:rx /opt/systemA
```

Listar direcotrios permisos rx

Seminario 3:

1. Seguir los pasos del seminario anterior sobre la máquina virtual base. Se debe entregar la salida de la ejecución de lanzar "ls -la /" dentro de la chroot jail.

```
root@ubuntu:/etc/systemd/system# chroot /example/chroot ls -la /
total 76
drwxr-xr-x  20 root root 4096 Jun 16 20:36 .
drwxr-xr-x  20 root root 4096 Jun 16 20:36 ..
drwxr-xr-x   2 root root 4096 Jun 16 20:36 bin
drwxr-xr-x   2 root root 4096 Apr 12  2016 boot
drwxr-xr-x   4 root root 4096 Jun 16 20:36 dev
drwxr-xr-x  42 root root 4096 Jun 16 20:47 etc
drwxr-xr-x   2 root root 4096 Apr 12  2016 home
drwxr-xr-x   8 root root 4096 Jun 16 20:36 lib
drwxr-xr-x   2 root root 4096 Jun 16 20:36 media
drwxr-xr-x   2 root root 4096 Jun 16 20:36 mnt
drwxr-xr-x   2 root root 4096 Jun 16 20:36 opt
drwxr-xr-x   2 root root 4096 Apr 12  2016 proc
dr-xr-xr-x 153 root root    0 Jun 16 20:17 root
drwxr-xr-x   4 root root 4096 Jun 16 20:36 run
drwxr-xr-x   2 root root 4096 Jun 16 20:36 sbin
drwxr-xr-x   2 root root 4096 Jun 16 20:36 srv
drwxr-xr-x   2 root root 4096 Feb  5  2016 sys
drwxrwxrwt   2 root root 4096 Jun 16 20:36 tmp
drwxr-xr-x  10 root root 4096 Jun 16 20:36 usr
drwxr-xr-x  11 root root 4096 Jun 16 20:36 var
```

2. Elaborar una lista de los principales comandos utilizados junto con una breve descripción de que hace cada uno.

Creamos instalación de Ubuntu en el path indicado: debootstrap

```
root@ubuntu:/home/ubuntu# debootstrap --variant=buildd --arch i386 xenial /example/chroot http://archive.ubuntu.com/ubuntu
```

Montar rutas accesibles desde la chrootjail: mount

```
root@ubuntu:/home/ubuntu# mount --bind /proc /example/chroot/root/
```

Entrar en la jaula:

```
root@ubuntu:/home/ubuntu# chroot /example/chroot
root@ubuntu:/# _
```

Lanzar comandos dentro de la jaula:

```
root@ubuntu:/home/ubuntu# chroot /example/chroot ls /
bin  boot  dev  etc  home  lib  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
```

Ejemplo de servicio en jaula:

```
[Unit]
Description=Chroot

[Service]
Type=simple
WorkingDirectory=/
ExecStart=/bin/bash -c "ls /"
RootDirectory=/example/chroot
User=root
Restart=on-failure
RestartSec=10

[Install]
WantedBy=multi-user.target
```

3. De la lista de cuestiones que se pueden encontrar en la sección 1.2, seleccionar aquellas que están relacionadas con una mala encapsulación en el sistema de ficheros.

ETAPA TEMPRANA

- ¿qué software expone?, ¿versiones?

- ¿qué sistema operativo?

- ¿qué software no expone, pero se presupone?

ETAPA INTERMEDIA

-¿software con vulnerabilidades conocidas?

- ¿qué recursos internos hay en la máquina?

- ¿cuándo parchearan el fallo?

ETAPA FINAL

-¿cómo puedo explotar los recursos?

- ¿necesito extraer recursos?

- ¿cómo puedo extraerlos?

- ¿cómo puedo acelerar la extracción?

- ¿puedo destruir recursos?

Seminario 4:

Crear los límites de trabajo según la especificación

Especificaciones:

• Crear en el sistema los usuarios iobound y cpubound.

Para el usuario iobound usamos adduser porque queremos que tenga un home, Shell, etc El usuario cpubound lo creamos con useradd (no queremos que tenga Shell, home, etc)

```
root@ubuntu:/home/ubuntu# adduser iobound
```

```
root@ubuntu:/home/ubuntu# useradd cpubound
```

• Limitar los procesos del usuario iobound a como máximo 1 minuto de uso de CPU.

Editamos el fichero de configuración de limites: nano /etc/security/limits.conf

Y añadimos:

```
iobound          hard    cpu             1
```

• Limitar el máximo número de procesos del usuario cpubound a 3.

Editamos el fichero de configuración de limites: nano /etc/security/limits.conf

Y añadimos:

```
cpubound         hard    nproc           3
```

• Limitar máximo número de ficheros abiertos global de la máquina a 1000.

Editamos el fichero de configuración de limites: nano /etc/security/limits.conf

Y añadimos:

```
*                hard    nofile          10000
```

• Establecer la prioridad por defecto -10 a iobound y 10 a cpubound.

Editamos el fichero de configuración de limites: nano /etc/security/limits.conf

Y añadimos:

```
iobound          -       priority        -10
cpubound         -       priority        10_
```

• Limitar el uso de disco de la unidad montada en /mnt/externalDisk a 5 ficheros y como máximo 1 bloque por fichero al usuario iobound.

Previa modificación del fichero /etc/fstab para permitir el uso de cuota, ejecutamos los siguientes comandos:

```
root@ubuntu:/mnt/externalDisk/home# setquota -u iobound 2048 3048 5 10 /mnt/externalDisk
```

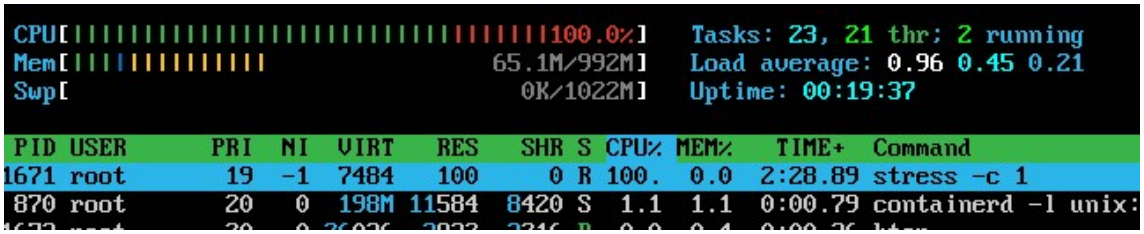
Usamos setquota, elegimos el usuario y le pasamos los parámetros (soft hard) correspondientes al número de bloques y el número de ficheros.

```
root@ubuntu:/mnt/externalDisk/home# quota iobound
Disk quotas for user iobound (uid 1008):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/sdb1       1    2048    3048               1       5      10
```

Estresar el sistema con stress y monitorizar con htop:

stress -c 1 & (Ejecutamos un worker en segundo plano)

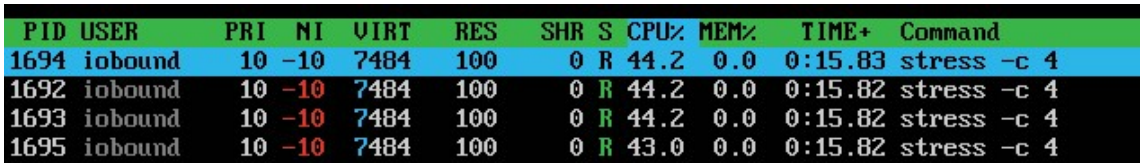
htop para monitorizar:



Comprobando los limites:

su iobound -c “stress -c 4” &

htop



Seminario 5:

Ejecutando procesos dentro de un docker

Especificaciones:

• Búsqueda de la imagen oficial de Ubuntu 16.04.

docker search Ubuntu

docker pull Ubuntu:16.04

```
root@ubuntu:/# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              16.04               065cf14a189c        31 hours ago        135 MB
```

• Creación de un contenedor a partir de la imagen base.

docker run -i -t 065 /bin/bash

```
root@ubuntu:/# docker images
REPOSITORY          TAG                 IMAGE ID
ubuntu              16.04               065cf14a189c
root@ubuntu:/# docker run -i -t 065 /bin/bash
root@44960c0182dc:/# _
```

• Personalización del contenedor para instalar el componente htop.

Actualizar repositorios: apt-get update

Instalar htop: apt-get install htop

htop

```
  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  220 root       20   0 22992  3428  2816 R  0.0  0.3  0:00.01 htop
    1 root       20   0 18232  3328  2856 S  0.0  0.3  0:00.01 /bin/bash
```

• Generación de una nueva imagen a partir del contenedor anterior.

Docker commit para generar la imagen a partir de nuestro contenedor:

```
root@ubuntu:/# docker ps
CONTAINER ID        IMAGE
PORTS               NAMES
44960c0182dc        065
                    practical_sinouss
root@ubuntu:/# docker commit 449
sha256:82d7713f771bc951b770203f559bf4
```

```
root@ubuntu:/# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>              82d7713f771b        30 seconds ago      166 MB
ubuntu              16.04               065cf14a189c        32 hours ago        135 MB
```

```
root@ubuntu:/# docker tag 82d dockerpruebamaster:1.0
root@ubuntu:/# docker images
REPOSITORY          TAG                 IMAGE ID
dockerpruebamaster  1.0                 82d7713f771b
ubuntu              16.04               065cf14a189c
```

• Detener el contenedor anterior.

Con el comando docker ps comprobamos que contenedores tenemos en marcha.

Posteriormente con docker stop “númeroID” paramos el contenedor. (No es necesario poner el id completo)

```
root@ubuntu:/home/ubuntu/docker/saludo# docker ps
CONTAINER ID        IMAGE               COMMAND
PORTS               NAMES
44960c0182dc        065                 "/bin/bash"
                    practical_sinoussi
root@ubuntu:/home/ubuntu/docker/saludo# docker stop 449
449
root@ubuntu:/home/ubuntu/docker/saludo# docker ps
CONTAINER ID        IMAGE               COMMAND
PORTS               NAMES
root@ubuntu:/home/ubuntu/docker/saludo# _
```

• Crear un nuevo contenedor a partir de la imagen anterior compartiendo la carpeta /home/ubuntu/saludo con la máquina anfitriona donde se escribirá un fichero de salida. El directorio compartido deberá tener permisos de lectura y escritura.

Para lanzar el script saludar.sh desde el contenedor tenemos que ejecutar el siguiente comando:

```
root@ubuntu:/# docker run --rm -i -t -v /home/ubuntu/docker/saludo:/saludo:rw -v /home/ubuntu/docker
//saludar.sh:/saludar.sh:ro dockerpruebamaster:1.0 /saludar.sh
```

Lo mas importante del mismo es el uso del parámetro -v (volumenes) en el cual le indicamos el volumen de la máquina donde queremos que lea y escriba el contenedor y también donde queremos que eso ocurra dentro del mismo:

-v /hom/ubuntu/docker/saludo:saludo:rw

Y lo mismo con el script:

/home/ubuntu/Docker/saludar.sh:/saludar.sh:ro en este caso con los permisos read only. Solo queremos que lo lea para que lo pueda ejecutar.

Al final del comando le indicamos que ejecute el script y podemos comprobar en nuestra máquina el resultado:

```
root@ubuntu:/home/ubuntu/docker/saludo# cat hola.txt
hola
hola
```

Como lo he ejecutado dos veces, el script ha escrito dos veces en el fichero de salida: hola.txt en nuestra máquina, fuera del contenedor.

Seminario 6:

Hay una aplicación que requiere que se permita el tráfico de entrada y salida por el puerto 9006, permitiendo conexiones con estado “ESTABLISHED” y “RELATED”. Lance los comandos necesarios.

```
root@ubuntu:/# iptables -A INPUT -p tcp -m tcp --sport 9006 -m state --state RELATED,ESTABLISHED -j
ACCEPT
root@ubuntu:/# iptables -A OUTPUT -p tcp -m tcp --dport 9006 -j ACCEPT
root@ubuntu:/#
```

Crear una clase HTB con ratio 256kbit y limitada a 1.5mbit. Su padre será “1:” y su identificador será “1:2”.

Añadir un filtro de tipo “u32” que asigne la clase “1:2” al tráfico con IP destino 10.10.10.10, con máscara de red 255.255.255.255.

```
root@ubuntu:/# tc class add dev enp0s17 parent 1: classid 1:2 htb rate 256kbit ceil 1.5mbit
```

```
root@ubuntu:/# tc filter add dev enp0s17 protocol ip parent 1: prio 1 u32 match ip dst 10.10.10.10/2
55.255.255.255 flowid 1:2
```