REPORT DI LABORATORIO: EXPLOITATION TELNET & METERPRETER UPGRADE

Studente: Rocco Paolo Caccamo

Corso: Cyber Security & Ethical Hacking (Epicode)

Modulo: S7 L2 - Network Security & Metasploit Framework **Data:** 20/01/2026

1. OBIETTIVO E SCENARIO

L'obiettivo dell'esercitazione è testare la sicurezza del protocollo Telnet su un sistema legacy, sfruttando credenziali di default per ottenere un accesso remoto e successivamente elevare la qualità della connessione trasformando una shell semplice in una sessione Meterpreter. Lo scenario prevede un attacco verso la macchina target **Metasploitable 2** (`192.168.1.149`), utilizzando la suite **Metasploit Framework** su Kali Linux.

2. FASE 1: RICOGNIZIONE E SCANSIONE (RECON)

Prima di tentare l'accesso, è stato necessario identificare con precisione il servizio attivo e la sua versione (Banner Grabbing). **Azioni eseguite:**

1. Ricerca dei moduli scanner per Telnet (`search type:auxiliary scanner telnet`).
2. Configurazione del modulo `auxiliary/scanner/telnet/telnet_version` impostando l'IP target (`RHOSTS 192.168.1.149`).

Eseguendo il modulo, ho ottenuto il banner `Metasploitable login:`, confermando che il sistema operativo è Linux e il servizio è attivo sulla porta 23.

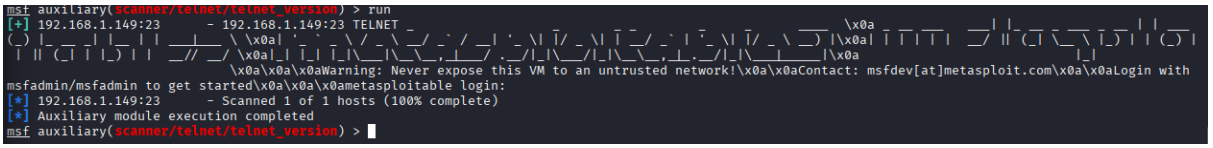

Fig 1: Scansione del servizio riuscita con identificazione del banner del sistema operativo.

3. FASE 2: ACCESSO INIZIALE (EXPLOITATION)

Una volta confermato il target, ho utilizzato il modulo `auxiliary/scanner/telnet/telnet_login` per effettuare un attacco mirato utilizzando le credenziali di default note per questo ambiente di laboratorio (`msfadmin:msfadmin`).

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

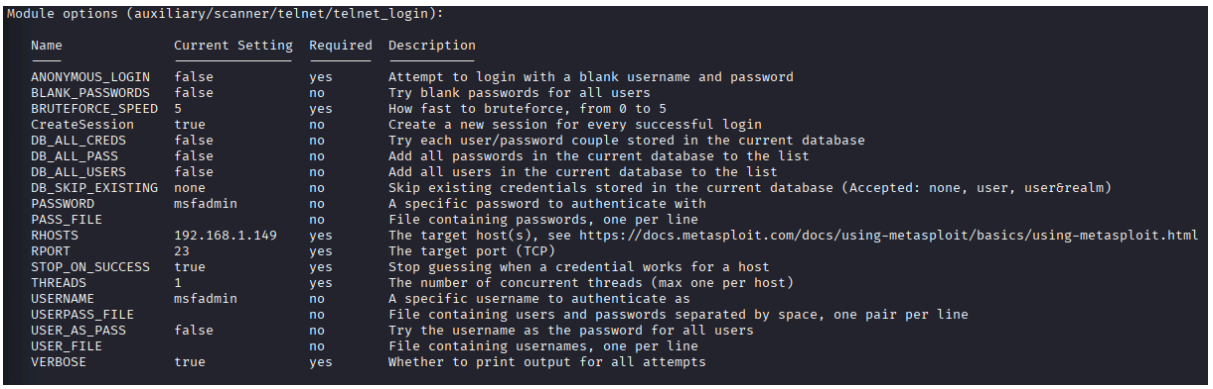

Fig 2: Configurazione dei parametri di attacco (Username, Password, Rhosts).

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

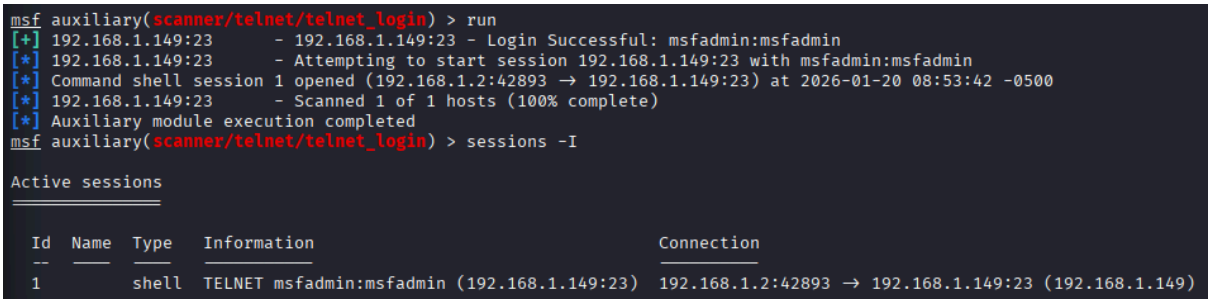

Fig 3: Login riuscito e apertura della sessione 1.

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.

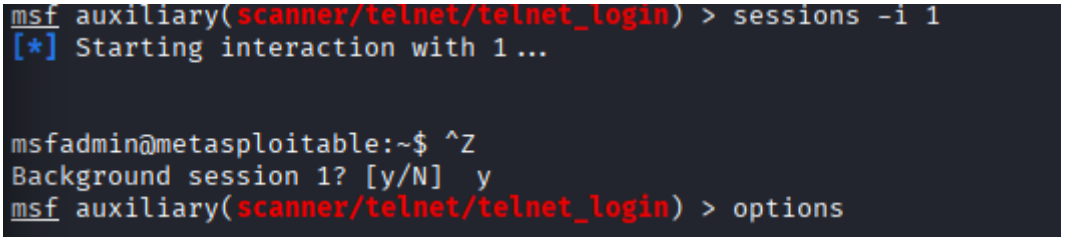

Fig 4: Interazione con la shell e messa in background.

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`.

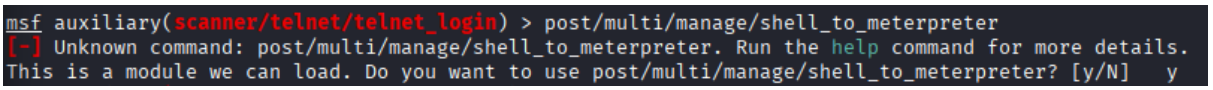

Fig 5: Errore di sintassi e correzione automatica suggerita dalla console.

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**.

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.