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:

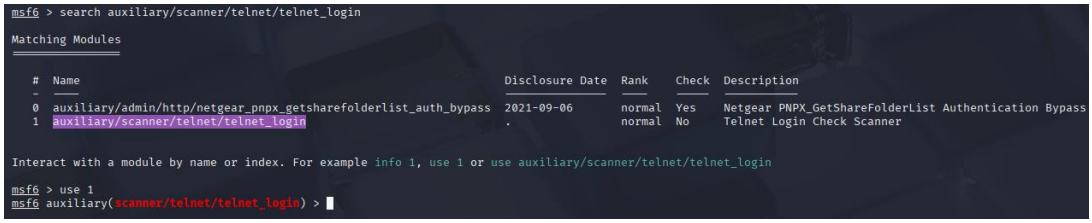
- search auxiliary/scanner/telnet/telnet_login

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.

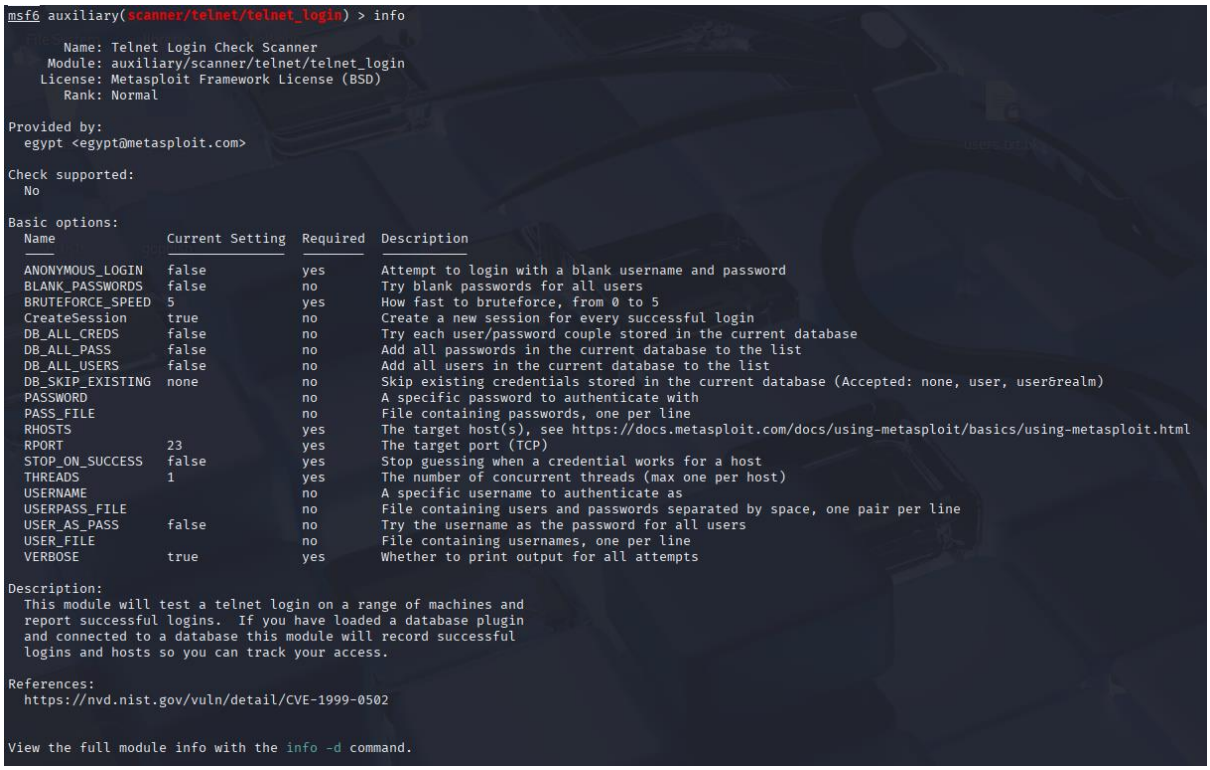


Fase 2 – Configurazione ed esecuzione del modulo Telnet Login Scanner

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).



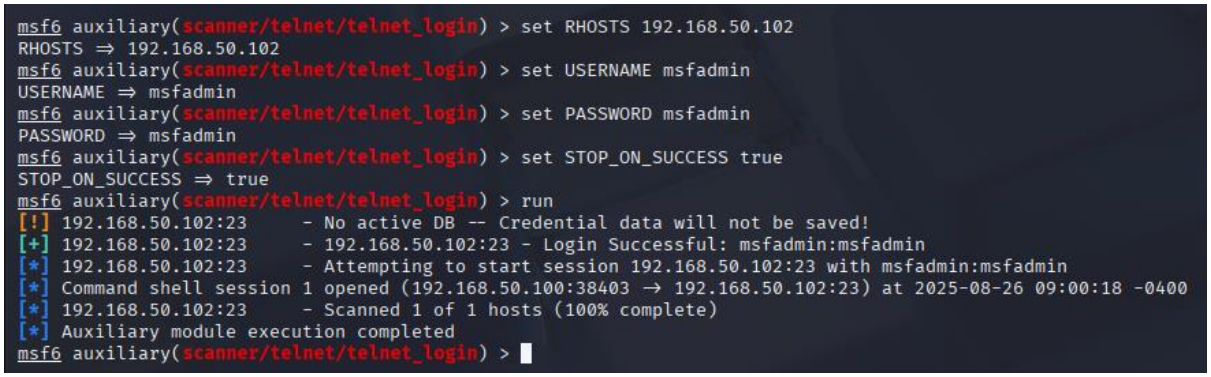
Dopo aver analizzato tutte le opzioni del modulo, ho configurato i parametri per iniziare il test. Ho scelto di puntare il modulo verso l'host `192.168.50.102`, impostando sia lo **username** che la **password** su `nsfadmin`. Si tratta di credenziali che sospettavo potessero essere valide, magari predefinite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



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

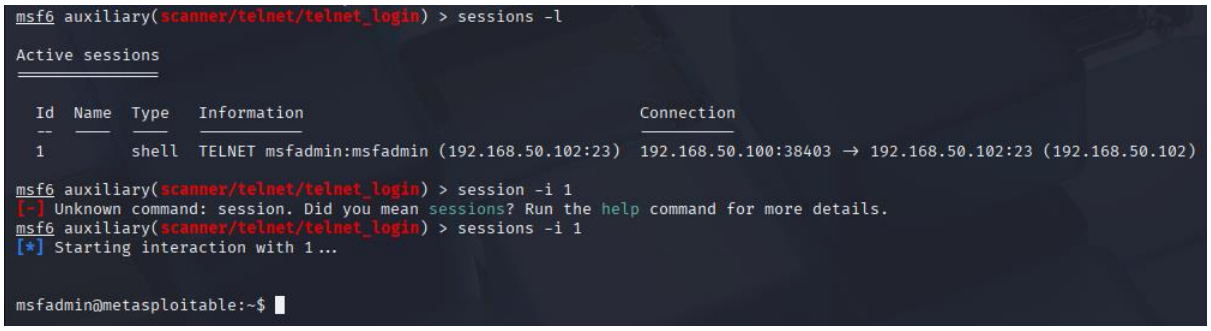
Il risultato mostra chiaramente una sessione aperta:

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:

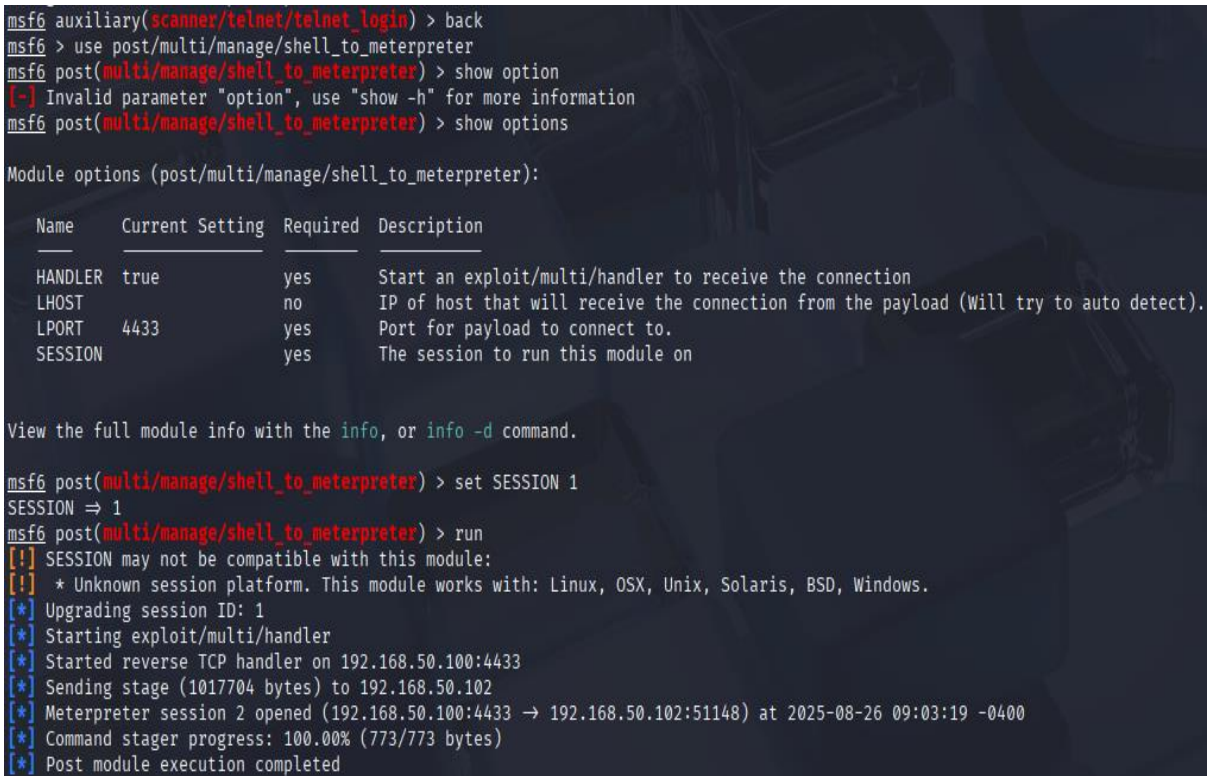
- sessions -i 1

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.

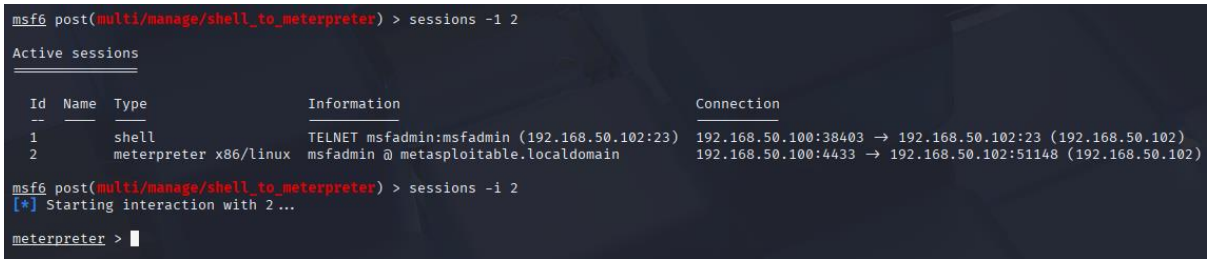


Fase 3 – Accesso avanzato con Meterpreter

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.



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.



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.